

UNIVERSIDADE FEDERAL DO PARANÁ

WLADIMIR GONÇALVES DE MORAIS

GERÊNCIA ATIVA DE FILAS PARA MELHORIA DA QUALIDADE NA TRANSMISSÃO
DE VÍDEO ADAPTATIVO

CURITIBA PR

2019

WLADIMIR GONÇALVES DE MORAIS

GERÊNCIA ATIVA DE FILAS PARA MELHORIA DA QUALIDADE NA TRANSMISSÃO
DE VÍDEO ADAPTATIVO

Dissertação apresentada ao curso de Pós-Graduação em Engenharia Elétrica, Área de Concentração Telecomunicações - Sistemas de Comunicação, Departamento de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Dr. Carlos Marcelo Pedroso.

CURITIBA PR

2019

Catálogo na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

M827g

Morais, Wladimir Gonçalves de

Gerência ativa de filas para melhoria da qualidade na transmissão de vídeo adaptativo [recurso eletrônico] /Wladimir Gonçalves de Moraes. – Curitiba, 2019.

Dissertação – Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, 2019.

Orientador: Carlos Marcelo Pedroso.

1. Vídeo digital. 2. Transmissão em rede. 3. Roteadores (Redes de computadores). I. Universidade Federal do Paraná. II. Pedroso, Carlos Marcelo. III. Título.

CDD: 004.678

Bibliotecária: Vanusa Maciel CRB- 9/1928

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **WLADIMIR GONÇALVES DE MORAIS** intitulada: **Gerência Ativa de Filas para Melhoria da Qualidade na Transmissão de Vídeo Adaptativo**, sob orientação do Prof. Dr. CARLOS MARCELO PEDROSO, que após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 23 de Agosto de 2019.



CARLOS MARCELO PEDROSO

Presidente da Banca Examinadora (UNIVERSIDADE FEDERAL DO PARANÁ)



MAURO SERGIO PEREIRA FONSECA

Avaliador Externo (UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ)



EVELIO MARTÍN GARCÍA FERNÁNDEZ

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)



LUIS HENRIQUE ASSUMPÇÃO LOLIS

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

O conteúdo de vídeo muito em breve representará 82% do tráfego mundial. Quando fluxos de vídeo concorrem por recursos de largura de banda no mesmo enlace, pode ocorrer o congestionamento, que por consequência provoca o aumento do atraso médio, *jitter* e taxa de perda de pacotes, impactando diretamente na qualidade de vídeo percebida pelo usuário. A ampliação superestimada da capacidade de armazenamento (*buffers*) de pacotes nos roteadores pode provocar o fenômeno conhecido por *bufferbloat*, que resulta em efeitos indesejados para o tráfego de vídeo ao vivo, devido ao aumento significativo da latência. Soluções de transmissão de vídeo adaptativo buscam minimizar os efeitos negativos do congestionamento através de diferentes representações do conteúdo em qualidades distintas. O DASH (*Dynamic Adaptive Streaming Over HTTP*) possibilita a transmissão de forma adaptativa e dinâmica, através da flexibilidade de transferências consecutivas de segmentos de vídeo de curta duração em diferentes qualidades, selecionando a melhor qualidade conforme condições instantâneas da rede. AQM (*Active Queue Management*) constitui uma solução complementar para o controle de congestionamento na camada de rede através de técnicas eficientes para gerência das filas de pacotes nos roteadores. Métodos AQM atuam de forma preventiva por meio do envio de informações para as fontes geradoras de tráfego sobre a iminência de um congestionamento na rede. O presente trabalho apresenta uma proposta de método AQM que realiza descartes aleatórios, monitorando o tempo de permanência dos pacotes na fila, com o objetivo de melhorar a qualidade na transmissão de conteúdo de vídeo ao vivo. A eficiência do método proposto é demonstrada a partir de simulações computacionais utilizando transmissão real de vídeo DASH com conteúdo H.264/AVC (*Advanced Video Coding*). A transmissão dos fluxos ocorre através de servidor e cliente real, sobre uma rede IP implementada no simulador NS-3. O fluxo DASH compete por recursos com outros fluxos no mesmo enlace de gargalo provocando descartes de pacotes na fila do roteador principal. O nível de utilização do enlace principal é intensificado gradativamente buscando reproduzir os efeitos do congestionamento. O método AQM é implementado na fila do roteador principal, sendo avaliado em comparação à outros métodos conhecidos. A avaliação da qualidade é realizada através do cálculo do PSNR (*Peak Signal Noise Ratio*) médio do vídeo recebido em comparação com o vídeo original, demonstrando a qualidade percebida pelo usuário em função do nível de utilização do enlace principal. Os resultados também avaliam o atraso médio e o *jitter* dos fluxos. Nos cenários simulados, na medida em que o congestionamento na rede se agrava, o método proposto resulta em maior PSNR, menor atraso médio e menor *jitter*, em comparação aos outros métodos avaliados.

Palavras-chave: AQM, Multimídia, Video Streaming, DASH.

ABSTRACT

Video content will soon represent 82% of global traffic. When video streams compete for bandwidth resources on the same link, congestion that consequently causes the increase of the average delay, jitter and rate of loss of packages, which directly impacts the quality of video perceived by the user. Overestimated packet buffer overflow on routers can cause the phenomenon known as bufferbloat that results in undesired effects for live video traffic due to the significant increase in latency. Adaptive video transmission solutions seek to minimize the negative effects of congestion through different representations of content in distinct qualities and may vary during transmission according to instant network conditions. The Dynamic Adaptive Streaming Over HTTP (DASH) enables the transmission of adaptive and dynamic, through the flexibility of consecutive transfers of short video segments with different qualities available. AQM (Active Queue Management) is a complementary solution for congestion control at the network layer, providing more efficient techniques for managing packet queues in the buffer space of the routers. AQM methods act in a preventive way by sending information to the traffic generating sources about the imminence of a congestion in the network. The present work presents a proposal of AQM method that performs random discarding, monitoring the queue time of the packages in order to improve the quality in the transmission of live video content. The efficiency of the proposed method is demonstrated from computational simulations using real DASH video transmission and H.264 / AVC (Advanced Video Coding) video content. The transmission of live streams occurs through server and actual client, over an IP network implemented in the NS-3 simulator. The DASH stream competes for resources with other streams on the same bottleneck link causing packet drops in the main router queue. The level of utilization of the main enclosure is intensified gradually to reproduce the effects congestion on the bottleneck link. The AQM method is implemented in the router queue compared to other known methods. Quality evaluation is performed by calculating the average Peak Signal Noise Ratio (PSNR) of the video received at comparison with the original video, demonstrating the quality perceived by the user in function of the level of use of the main link. The results also assess the average delay and jitter of flows. In the simulated scenarios, as link utilization increases, the proposed AQM method presented better average PSNR, lower delay and jitter, compared with other evaluated methods.

Keywords: AQM, Multimedia, Video Streaming, DASH.

LISTA DE FIGURAS

1.1	Impacto da perda de pacotes na qualidade de imagem.	12
1.2	Impacto da perda de pacotes na transmissão de vídeo.	13
1.3	Impacto da perda de pacotes na transmissão <i>Live Streaming</i>	14
1.4	Esquema simplificado de ocorrência do fenômeno <i>Bufferbloat</i>	15
2.1	Principais etapas envolvidas na realização do estudo.	17
2.2	Cenário utilizado para avaliação de desempenho.	18
3.1	Representação do efeito da sincronização global das fontes TCP.	22
3.2	Esquema de integração entre os principais componentes utilizados pelo método AQM PIE.	25
3.3	Sequência de quadros que formam o vídeo <i>Football</i>	26
3.4	Estrutura de um GoP (9,2).	27
3.5	Quadro do vídeo <i>Sunflower</i> em 2 diferentes representações de qualidade.	29
3.6	Comportamento adaptativo do DASH ao longo do tempo.	30
4.1	Transição entre segmentos durante a transmissão do vídeo <i>Big Buck Bunny</i> com o DropTail.	36
5.1	Os vídeos, em sentido horário, <i>Sunflower</i> , <i>Touchdown Pass</i> , <i>Big Buck Bunny</i> , <i>Pedestrian Area</i> , <i>Riverbed</i> e <i>Rush Hour</i>	39
5.2	Esquema de operação do módulo <i>Tap Bridge Network Device</i> no simulador NS-3.	42
5.3	Configuração para análise de desempenho dos métodos AQM.	43
6.1	PSNR médio resultante para o vídeo <i>Big Buck Bunny</i>	49
6.2	PSNR médio resultante para o vídeo <i>Rush Hour</i>	50
6.3	PSNR médio resultante para o vídeo <i>Sunflower</i>	50
6.4	PSNR médio resultante para o vídeo <i>Touchdown Pass</i>	51
6.5	PSNR médio resultante para o vídeo <i>Pedestrian Area</i>	52
6.6	PSNR médio resultante para o vídeo <i>Riverbed</i>	52
6.7	Atraso médio resultante para o vídeo <i>Big Buck Bunny</i>	53
6.8	Jitter resultante para o vídeo <i>Big Buck Bunny</i>	54
6.9	Atraso médio resultante para o vídeo <i>Rush Hour</i>	54
6.10	Jitter resultante para o vídeo <i>Rush Hour</i>	55
6.11	Atraso médio resultante para o vídeo <i>Sunflower</i>	55
6.12	Jitter resultante para o vídeo <i>Sunflower</i>	56
6.13	Atraso médio resultante para o vídeo <i>Touchdown Pass</i>	56

6.14	Jitter resultante para o vídeo <i>Touchdown Pass</i>	57
6.15	Atraso médio resultante para o vídeo <i>Pedestrian Area</i>	57
6.16	Jitter resultante para o vídeo <i>Pedestrian Area</i>	58
6.17	Atraso médio resultante para o vídeo <i>Riverbed</i>	58
6.18	Jitter resultante para o vídeo <i>Riverbed</i>	59
6.19	Transição entre os segmentos de vídeo solicitados pelo cliente para o vídeo <i>Big Buck Bunny</i> no nível de ocupação de 95%.	61

LISTA DE TABELAS

3.1	Comparação entre os métodos AQM discutidos.	26
5.1	Principais Recursos Utilizados para Avaliação de Desempenho	38
5.2	Parâmetros dos vídeos utilizados na avaliação de desempenho.	39
5.3	Características técnicas de imagem e movimento dos vídeos.	40
5.4	Configuração dos Métodos Avaliados.	46
5.5	Relação entre PSNR e MOS.	47
5.6	PSNR médio Máximo - Padrão Ouro.	47
6.1	Índice de justiça de <i>Jain</i> para o <i>Throughput</i>	60
6.2	Índice de justiça de <i>Jain</i> para o atraso médio.	60

LISTA DE ACRÔNIMOS

ABS	Adaptive Bitrate Streaming
AQM	Active Queue Management
AVC	Advanced Video Coding
CDN	Content Delivery Network
DASH	Dynamic Adaptive Streaming over HTTP
DASH-IF	DASH Industry Forum
EWMA	Exponentially Weighted Moving Average
GOP	Group-of-Picture
HD	High-Definition
HTTP	Hypertext Transfer Protocol
IPTV	Internet Protocol television
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU	International Telecommunication Union
MOS	Mean Opinion Score
MPD	Media Presentation Description
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
PPGEE	Programa de Pós-Graduação em Engenharia Elétrica
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality-of-Experience
QoS	Quality-of-Service
RTT	Round Trip Time
SSIM	Structural SIMilarity
SVC	Scalable Video Coding
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UFPR	Universidade Federal do Paraná
URL	Uniform Resource Locator
VBR	Variable Bitrate
VoD	Video-on-Demand

SUMÁRIO

1	Introdução	11
1.1	Tráfego de Vídeo na Internet	11
1.1.1	Estatísticas e Previsões	11
1.2	Delimitação do Problema	12
1.2.1	O Fenômeno <i>Bufferbloat</i>	14
1.3	Organização do Trabalho	15
2	Objetivos e Metodologia	16
2.1	Objetivos do Estudo	16
2.2	Metodologia Utilizada	16
3	Conceitos Fundamentais	19
3.1	Gerência Ativa de Filas	19
3.1.1	DropTail	19
3.2	Métodos AQM	20
3.2.1	RED	20
3.2.2	ARED	22
3.2.3	CoDel	23
3.2.4	PIE	24
3.2.5	Análise Comparativa entre os Métodos	26
3.3	Codificação de Vídeo	26
3.3.1	<i>Advanced Video Coding</i> - H.264/AVC	27
3.4	Transmissão de Vídeo pela Internet	28
3.4.1	<i>Dynamic Adaptive Streaming over HTTP</i> - DASH	29
3.4.2	DASH e o Estado da Arte	30
3.5	Aplicações de Métodos AQM	31
3.5.1	Melhoria da qualidade de <i>Video Streaming</i> com AQM	31
4	Método Proposto	33
4.1	Paradigma do Modelo	33
4.2	Parâmetros e Algoritmo	33
4.3	Benefícios Esperados	35
5	Avaliação de Desempenho	37
5.1	Cenário de Avaliação	37
5.1.1	Conteúdo de Vídeo	38

5.1.2	Ambiente Computacional	41
5.2	Implementação das Simulações	42
5.2.1	Definições de Rede	42
5.2.2	Tráfego Concorrente	43
5.2.3	Dinâmica dos Experimentos	44
5.2.4	Parametrização dos Métodos AQM	45
5.3	Análises de Resultados	45
5.3.1	<i>Peak Signal to Noise Ratio</i> - PSNR	45
5.3.2	Padrão Ouro	47
5.3.3	Atraso Médio e <i>Jitter</i>	47
6	Resultados	48
6.1	PSNR Médio	48
6.2	Atraso Médio e <i>Jitter</i>	53
6.3	Justiça entre Fluxos	59
6.4	Transição entre Qualidades	60
6.5	Discussão dos Resultados	62
7	Conclusão	64
7.1	Trabalhos Futuros	65
	Referências	66

Capítulo 1

Introdução

1.1 Tráfego de Vídeo na Internet

O tráfego de vídeo é responsável por uma grande porção dos dados transmitidos nas redes, quando comparado com os outros tipos de mídia [1]. Provedores de conteúdo como *Youtube* e *Netflix* possuem milhões de usuários em todo o mundo. O fluxo de vídeo codificado possui um comportamento de taxa variável (VBR - *Variable Bit Rate*). Com o aumento do número de dispositivos com acesso à Internet, as aplicações de *streaming* acabam concorrendo pela largura de banda da última milha da rede de uma ISP (*Internet Service Provider*), criando gargalos que afetam a qualidade do serviço [2].

Dependendo do nível de utilização das filas dos roteadores, esse comportamento pode ocasionar congestionamentos, levando a possíveis perdas de pacotes e variação do atraso de propagação na rede. Por esse motivo, tem-se, em determinados momentos de pico, uma enorme disputa pela utilização dos recursos disponíveis, tais como a largura de banda dos enlaces de telecomunicações e capacidade de processamento dos ativos de rede envolvidos.

Na Internet o tráfego de vídeo encontra-se exposto às consequências negativas derivadas do congestionamento, que resultam na ausência de garantia de vazão, aumento do atraso e *jitter* [3], além de um aumento significativo na probabilidade de perda de pacotes. Com o congestionamento, frações de pacotes acabam sendo descartadas, reduzindo a qualidade de vídeo percebida pelo usuário, limitando a continuidade dos serviços ou até mesmo inviabilizando o funcionamento de aplicações de *streaming* de vídeo.

1.1.1 Estatísticas e Previsões

Segundo um recente relatório de uma das maiores fabricantes de equipamentos de telecomunicações, o conteúdo de vídeo foi o responsável por mais de 73% de todo o tráfego da Internet do mundo em 2016, e muito em breve a transmissão de vídeo representará 82% do tráfego mundial [4].

O *streaming* de vídeo, por se tratar de conteúdo em tempo real, é composto de rajadas de pacotes que necessitam de garantia de entrega no destino dentro de um limite temporal restrito, pois se trata de informação sensível ao atraso, ao *jitter* e à perda de pacotes. Portanto esse tipo de tráfego exige tratamento diferenciado para o encaminhamento dos pacotes, pois o protocolo IP (*Internet Protocol*), em sua concepção original, não contempla o controle de fluxo e de congestionamento, ficando este a cargo de protocolos das camadas superiores, por exemplo, o protocolo TCP (*Transport Control Protocol*). O relatório *Cisco Forecast and Trends 2017–2022* [4] ainda destaca algumas previsões e tendências para os próximos anos, dentre as quais pode-se citar:

- "Em 2021, um único indivíduo levaria mais de 5 milhões de anos para assistir à quantidade de vídeo trafegada na Internet a cada segundo."
- "No ano de 2021, um milhão de minutos de conteúdo de vídeo trafegará pela rede global a cada segundo."

Diante do exposto, percebe-se que o cenário de transmissão de vídeo pela Internet atingirá níveis bastante elevados, o que demandará a implantação de estratégias computacionais que visem uma forma de utilização mais eficiente dos recursos e equipamentos que constituem os enlaces das operadoras de telecomunicações, principalmente nas redes de acesso.

Dentre as aplicações de vídeo, nota-se que a demanda de maior impacto está relacionada ao consumo de vídeos de longa duração, ou seja, filmes e séries dos mais variados gêneros, que são disponibilizados por provedores de conteúdo e serviços de TV por internet, dos quais destacam-se o *Netflix*, *Youtube* e *Amazon*, devido ao enorme número de usuários por todo mundo [5]. O tráfego de vídeo sob demanda de longa duração (*Long-Form Internet VoD*) já ocupa a maior parcela de consumo na rede e continuará crescendo até o ano de 2021 [4]. Para agravar ainda mais o problema, nos próximos anos haverá um forte incremento na qualidade dos vídeos transmitidos, como por exemplo a tecnologia 4K [4], o que irá pressionar ainda mais as infraestruturas de rede disponíveis.

1.2 Delimitação do Problema

Ao assistir um vídeo pela Internet, o usuário pode ter sua expectativa frustrada caso a qualidade da imagem deixe a desejar, fazendo com que a experiência do expectador seja ruim ao perceber a demora para reprodução do conteúdo por meio de interrupções na transmissão e visualização de imagens defeituosas. Como resultado do aumento do atraso e da taxa de perdas, o vídeo pode apresentar diversas falhas, conhecidas por artefatos, fantasmas, travamentos, congelamentos que provocam indefinições na composição das imagens, podendo ocasionar até mesmo a total incompreensão visual do conteúdo recebido [6]. A figura 1.1 apresenta o resultado da perda de pacotes na constituição da qualidade da imagem no destino com quadro do vídeo *Sunflower* [7].

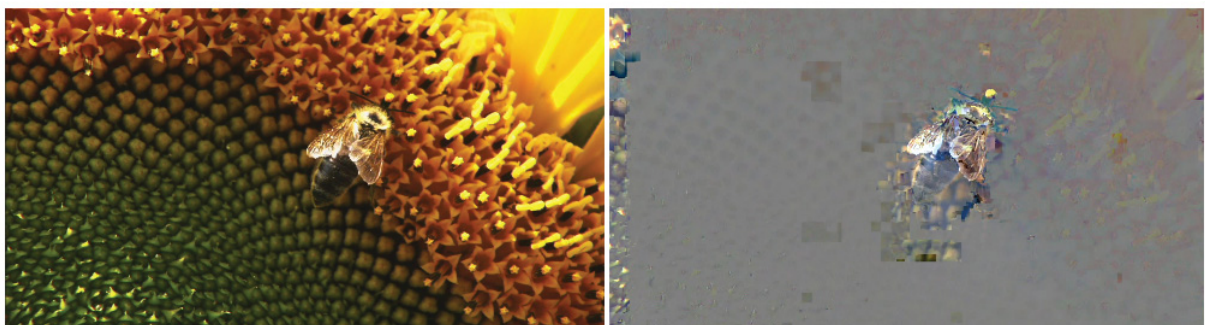


Figura 1.1: Impacto da perda de pacotes na qualidade de imagem.

Em certos casos as redes IP (Internet Protocol) não são capazes de oferecer níveis de qualidade de serviço adequados aos requerimentos necessários para fluxos de vídeo [8]. Apesar da inadequação da especificação original do protocolo *IP* à transmissão de fluxos multimídia em tempo-real, diversas especificações foram realizadas para adequar o protocolo às necessidades atuais. Originalmente o *IP* suportava apenas transmissão de dados através do

melhor esforço (*Best Effort*). No fim da década de 1990 foram especificados mecanismos de qualidade de serviço (*Quality of Service - QoS*), conhecidos por Serviços Integrados (*IntServ*) e Serviços Diferenciados (*DiffServ*), que estenderam as capacidades originais, estando amplamente disponíveis nos dispositivos atuais na Internet. No entanto, em grande parte, o núcleo da Internet permanece ainda operando com o conceito de melhor esforço.

Algoritmos de codificação de vídeo como o H.264/AVC (*Advanced Video Coding - AVC*) são capazes de fornecer boa qualidade de vídeo com baixas taxas de *bits* por segundo, possibilitando alta resolução. Por sua vez, as tecnologias de transmissão de vídeo adaptativo, como o DASH (*Dynamic Adaptive Streaming over HTTP*) [9] surgiram com o propósito de atenuar os efeitos do congestionamento, fornecendo flexibilidade ao disponibilizar pequenos segmentos de vídeo em diferentes qualidades.

Para transmissões em tempo real, minimizar parâmetros como atraso, *Jitter* e descarte de pacotes é fundamental para manter a qualidade em níveis satisfatórios. Na ocorrência de congestionamentos, a qualidade pode ser afetada por períodos de descartes de pacotes e atrasos na entrega dos segmentos, o que provoca degradações na qualidade de vídeo percebida.

A Figura 1.2 apresenta um exemplo do impacto provocado pelo aumento do atraso e da taxa de perdas de pacotes em aplicações de vídeo, no qual os quadros do vídeo apresentam falhas conhecidas por artefatos e fantasmas, o que consequentemente resulta em indefinições na composição das imagens degradando bastante a qualidade percebida pelo usuário.

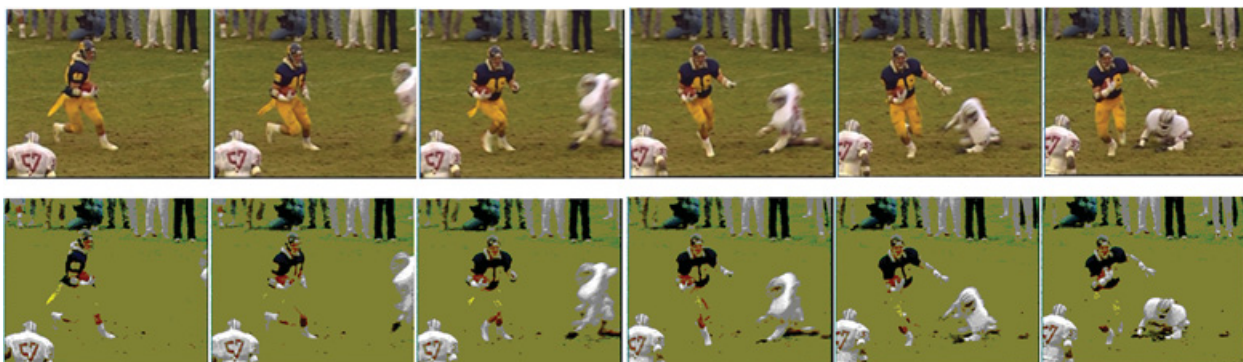


Figura 1.2: Impacto da perda de pacotes na transmissão de vídeo.

Em seguida, a Figura 1.3 representa uma falha bastante comum resultante do impacto provocado pelo aumento do atraso e da taxa de perdas de pacotes em aplicações de (*Live Streaming*), o que gera travamentos e/ou congelamentos durante a reprodução do conteúdo de vídeo que está sendo transmitido ao vivo, produzindo o efeito da demora na apresentação de novas imagens, após um tempo de espera por novos quadros, em que um único quadro estático é mostrado na tela durante um certo período de tempo. O efeito da latência em evento esportivo transmitidos ao vivo, podem frustrar os usuários, que aguardam um tempo maior para visualizar as imagens de cenas que já ocorreram, não refletindo de forma satisfatória o acompanhamento em tempo real da alternância de pontos e/ou placar do jogo.

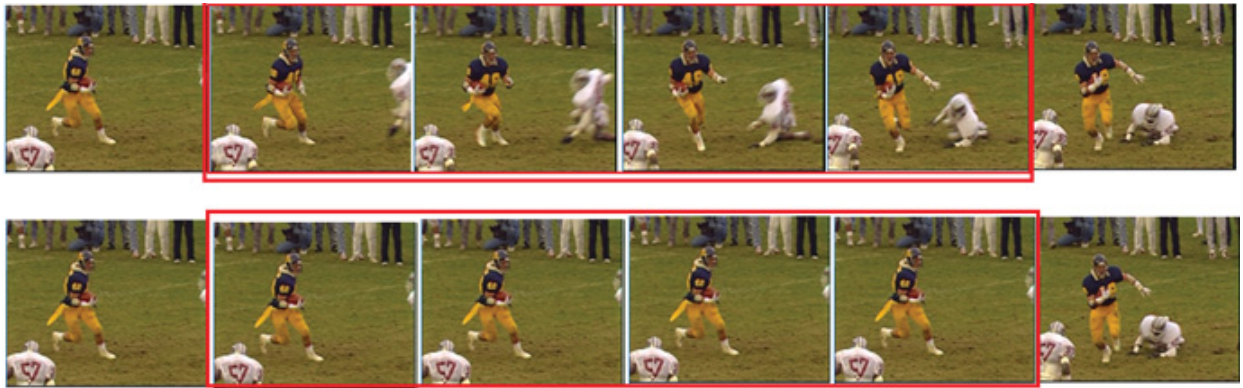


Figura 1.3: Impacto da perda de pacotes na transmissão *Live Streaming*.

1.2.1 O Fenômeno *Bufferbloat*

Os equipamentos intermediários da rede precisam ter alguma quantidade espaço para armazenar filas de pacotes, a fim de lidar com rajadas que chegam por um enlace de entrada e, para em seguida, poder realizar o encaminhamento para um enlace de saída.

Como alternativa para reduzir o descarte de pacotes, principalmente na última milha, onde tem-se gargalos na rede, as operadoras buscam aumentar o tamanho dos *buffers* dos roteadores, na tentativa de melhor acomodar o tráfego. Esta tendência foi impulsionada pela queda do preço das memórias [10]. Como consequência do enorme tamanho das filas de pacotes, pode ocorrer grande impacto na qualidade percebida pelo usuário que assiste à uma transmissão de vídeo ao vivo, devido principalmente à grande latência na entrega dos pacotes. Este fenômeno é conhecido por *Bufferbloat* [11] e resulta em efeitos indesejados para o tráfego de vídeo, pois os pacotes permanecem na fila por um tempo além do limite tolerado pelo cliente, tornando-se pacotes inválidos ao chegar no destino, devido ao esgotamento do tempo de reprodução dos mesmos no *player*.

Uma solução eficiente para o problema do *Bufferbloat* requer métodos de gerenciamento de filas que possibilitem a detecção antecipada do excesso de enfileiramento de pacotes e medidas para que a fila diminua o nível de ocupação a partir de descartes antecipados de pacotes. A Figura 1.4 ilustra a ocorrência do fenômeno *Bufferbloat* em que os pacotes chegam em rajadas pelo enlace de entrada, elevando o nível de utilização do *buffer* em sua capacidade máxima, promovendo alta latência na transmissão no enlace de saída.

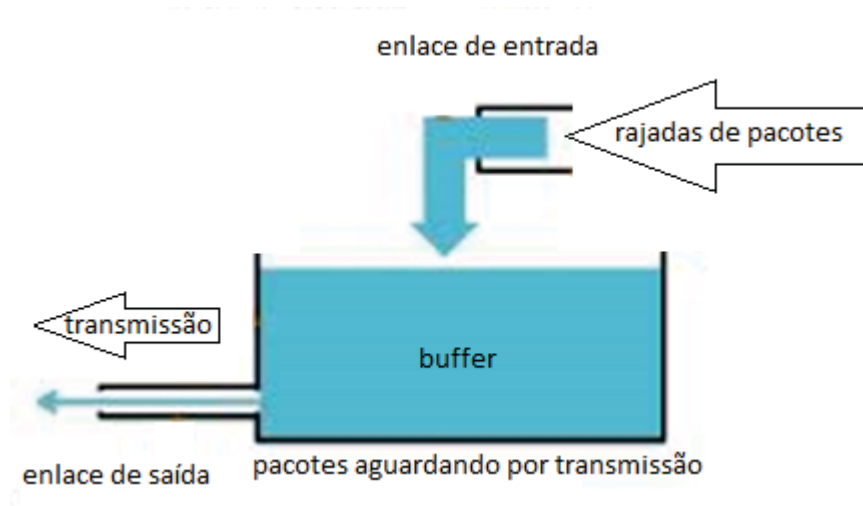


Figura 1.4: Esquema simplificado de ocorrência do fenômeno *Bufferbloat*.

Mecanismos controle de congestionamentos são cruciais para aplicações de *streaming* [12]. De maneira a atenuar os efeitos do *bufferbloat*, métodos de AQM [13] surgem como importante solução para prevenção de congestionamentos. Implementado nos roteadores, o AQM tem por objetivo prevenir o colapso total da rede, normalmente explorando a capacidade dos protocolos de camada 4 [8], notificando as fontes de transmissão sobre a iminência do congestionamento.

1.3 Organização do Trabalho

O presente trabalho encontra-se estruturado em 7 capítulos conforme descrito a seguir:

- O Capítulo 2 apresenta os objetivos e metodologia utilizada no estudo.
- O Capítulo 3 apresenta o contexto teórico e os conceitos fundamentais para compreensão do tema abordado, como os princípios básicos do Gerenciamento Ativos de Filas e da tecnologia de transmissão de vídeo com adaptabilidade dinâmica MPEG-DASH, além de apresentar alguns trabalhos relacionados, destacando-se algumas pesquisas de relevância na área;
- O Capítulo 4 descreve o modelo proposto e suas características básicas de operação que visam proporcionar a melhoria da qualidade de vídeo percebida pelo usuário;
- O Capítulo 5 apresenta a avaliação de desempenho bem como os experimentos que foram realizados para validação do modelo proposto, descrevendo as etapas para configuração do ambiente computacional e a implementação das simulações;
- O Capítulo 6 expõe os resultados obtidos nas simulações realizadas que demonstram a vantagem observada pelo modelo proposto;
- O Capítulo 7 apresenta as conclusões obtidas diante dos resultados observados em relação às expectativas atingidas além de elencar algumas propostas para trabalhos futuros.

Ao final do presente trabalho estão enumeradas as principais referências bibliográficas que foram utilizadas ao longo do estudo.

Capítulo 2

Objetivos e Metodologia

2.1 Objetivos do Estudo

O Objetivo Geral do trabalho é propor, implementar e avaliar um método AQM que possibilite a melhoria da qualidade de vídeo percebida pelo usuário na transmissão de vídeo adaptativo.

Os Objetivos Específicos correspondem aos seguintes:

- Desenvolver uma proposta de algoritmo para AQM por meio de alterações no algoritmo RED (*Random Early Detection*) [14], acrescentando funcionalidades de limites temporais para efetuar o descarte randômico de pacotes, visando a melhoria na qualidade em transmissão de *streaming* de vídeo ao vivo com a tecnologia DASH e conteúdo de vídeo codificado em H.264/AVC;
- Implementar através de simulações computacionais cenários de transmissão real de fluxo adaptativo de vídeo em uma rede IP, na qual o fluxo de vídeo adaptativo irá competir por recursos com outros fluxos concorrentes;
- Realizar a avaliação de desempenho do método AQM proposto, através da instalação no roteador que recebe todos os fluxos, comparando seu desempenho com outros modelos conhecidos;
- Avaliar quantitativamente e qualitativamente o impacto do uso do método AQM proposto, utilizando métricas objetivas como PSNR (*Peak Signal-to-Noise Ratio*) para avaliar o benefício obtido na qualidade percebida pelo usuário;
- Realizar a análise de outros parâmetros como atraso médio e o *jitter* na transmissão dos pacotes para avaliar o ganho obtido com a redução média da latência e variação da mesma durante a transmissão dos fluxos de vídeo.

2.2 Metodologia Utilizada

A metodologia utilizada no presente trabalho consiste na codificação de diferentes vídeos no padrão H.264/AVC, gerando segmentos de diversas qualidades para proporcionar adaptabilidade dinâmica, e transmiti-los por meio da tecnologia DASH. A transmissão dos vídeos foi realizada em um ambiente que combina servidor e cliente DASH reais com uma rede simulada que implementa uma topologia *dumbbell* [15] e elementos gerando tráfego de fundo.

Foram configurados cenários realistas de transmissão de vídeo ao vivo em uma rede contendo diferentes níveis de utilização do enlace principal. Durante a transmissão pelo servidor, o vídeo recebido pelo cliente foi armazenado, e posteriormente comparado com o vídeo original por meio da técnica PSNR.

No decorrer da simulação foram coletadas algumas estatísticas de tráfego, como atraso médio e *jitter* para elaboração de outras análises de resultados.

A figura 2.1 ilustra algumas das etapas envolvidas para tornar possível a realização do presente estudo, contemplando a codificação do vídeo, preparação do conteúdo para ser transmitido, transmissão do *streaming* adaptativo pelo servidor, reprodução e armazenamento do vídeo recebido no cliente, decodificação do vídeo recebido, e posterior avaliação de qualidade percebida através da comparação do vídeo original com o vídeo transmitido.

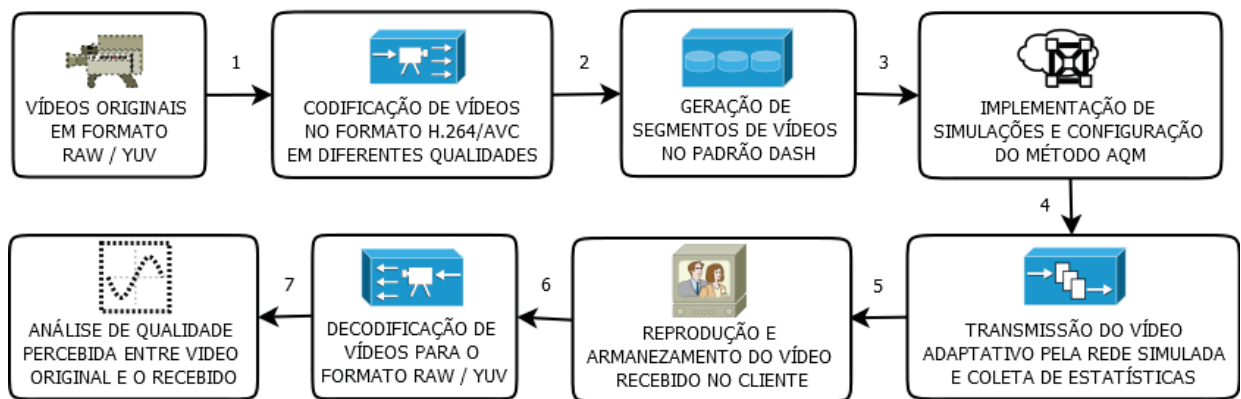


Figura 2.1: Principais etapas envolvidas na realização do estudo.

Na topologia simulada, os enlaces de acesso são muito rápidos e possuem um atraso insignificante, enquanto que o enlace principal possui uma baixa largura de banda e um atraso significativo. Sendo assim, o fluxo de vídeo adaptativo disputa recursos da rede com outros tipos de tráfego, reproduzindo os efeitos do congestionamento.

O método AQM é instalado na fila do roteador que recebe as fontes de tráfego no enlace principal. O roteador recebe os pacotes do servidor de vídeo e repassa ao cliente, além de também realizar o recebimento dos pacotes das fontes de tráfego concorrentes e repassá-los aos respectivos destinos. Diante dessa configuração, a competição por largura de banda no canal de gargalo gera descartes de pacotes na fila do roteador impactando diretamente na qualidade do vídeo recebido no cliente. A figura 2.2 apresenta de forma resumida a topologia necessária para ser realizada a avaliação de desempenho dos métodos de gerenciamento ativo de filas.

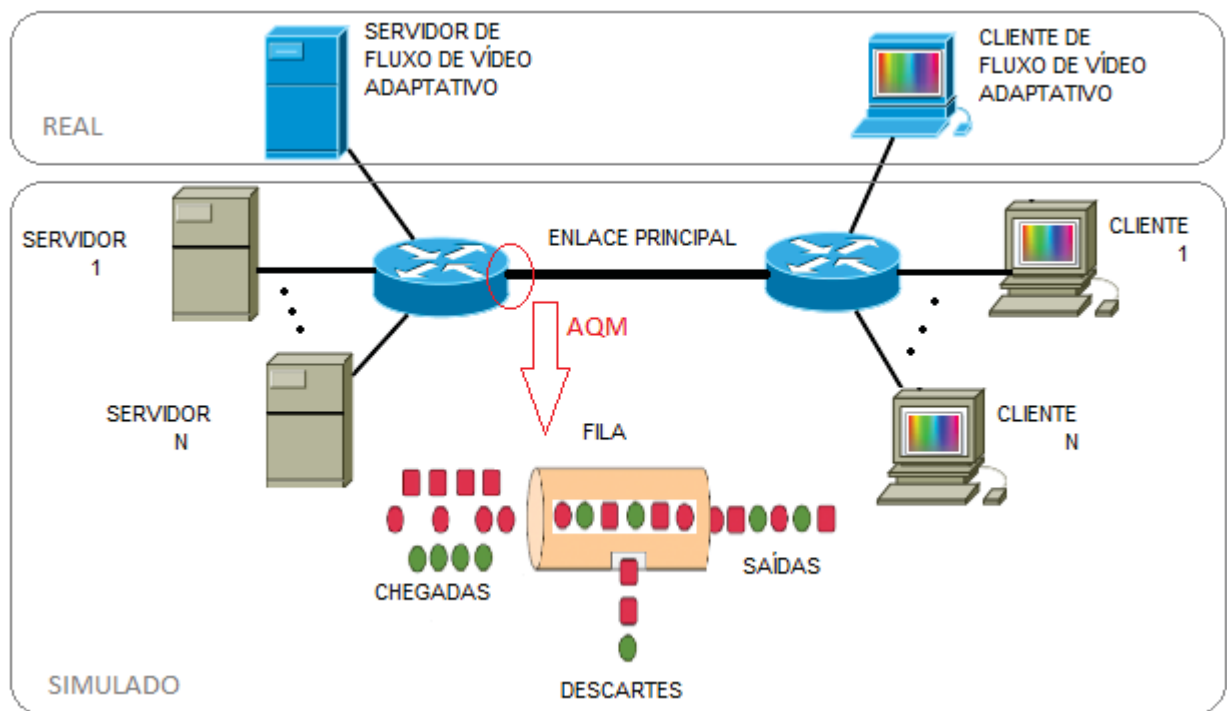


Figura 2.2: Cenário utilizado para avaliação de desempenho.

Capítulo 3

Conceitos Fundamentais

3.1 Gerência Ativa de Filas

O estudo do Gerenciamento Ativo de Filas, em inglês *Active Queue Management*, é mais conhecido pela sigla AQM e se relaciona diretamente com a teoria de filas, sendo um tema amplamente pesquisado e utilizado no campo da computação e telecomunicações [13]. Os algoritmos de AQM constituem-se de mecanismos controladores proativos de congestionamentos que atuam em um roteador [13], sendo responsáveis por detectar de forma antecipada tais eventos e notificar as fontes geradoras de tráfego. Tal notificação pode ser realizada por meio de descartes de pacotes ou através de uma marcação na extensão ECN (*Explicit Congestion Notification*) do protocolo IP, que possibilita a notificação das fontes de tráfego sobre o congestionamento sem a necessidade do descarte.

Por meio da identificação precoce do congestionamento, mecanismos de gerenciamento de filas buscam realizar o descarte antecipado de pacotes antes mesmo que seja esgotada a capacidade de receber novos pacotes na fila, evitando o colapso na rede. Dessa forma, os métodos AQM visam complementar o trabalho desempenhado pelos protocolos de transmissão de fim-a-fim, como o TCP (*Transport Control Protocol*), no controle e prevenção de congestionamento.

Os métodos AQM geralmente utilizam informações como o tamanho médio e instantâneo da fila, a taxa de chegada de pacotes e valores limites para fila cheia e vazia [13]. Porém, com o crescente conjunto de aplicações na rede IP, o foco dos algoritmos de AQM foi ampliado buscando colaborar no provisionamento de qualidade serviço, atendendo aos requerimentos dos diferentes tipos de fluxo de tráfego. Segundo R. Adams [13], os mecanismos de AQM são utilizados para trabalhar em cooperação com outros mecanismos de QoS (Quality of Service), tais como condicionadores de tráfego e escalonadores de pacotes de forma que combinados, seus efeitos possam distinguir diferentes tipos de tráfego, ofertando um melhor nível de serviço.

3.1.1 DropTail

Anteriormente aos esquemas de AQM, os roteadores geralmente utilizavam o algoritmo de descarte de pacotes conhecido por DropTail, o qual consiste em descartar pacotes somente quando a fila está com sua capacidade esgotada, deixando de fora do armazenamento todos os pacotes que chegam após o esgotamento do limite da fila até que a mesma diminua e tenha espaço para receber e enfileirar novos pacotes.

Sendo assim, com o método DropTail o descarte de pacotes é realizado de maneira ineficiente pois não leva em consideração o tipo de tráfego envolvido e nem faz justiça com as fontes que estão disputando o recurso, uma vez que pode descartar rajadas de informação de algumas fontes no momento em que a fila está saturada, e em outro momento, em que a fila esteja

vazia, aceitar uma boa quantidade de pacotes de outras fontes até que a capacidade da fila volte a atingir o seu limite [16].

Estudos, como [16], comprovam que descartar pacotes somente quando a fila está cheia não é uma boa estratégia para ser utilizada na Internet. Faz-se necessário o estabelecimento de critérios mais abrangentes e eficientes para a seleção e efetivação do descarte antecipado, possibilitando que as fontes de tráfego sejam informadas da iminência de um congestionamento e reduzam suas taxas de transmissão antes de que se esgote a capacidade de armazenamento nas filas dos *buffers* dos roteadores e a rede entre em colapso.

3.2 Métodos AQM

Através da adição de mecanismos de AQM na camada de rede, mais especificamente nos roteadores, é possível otimizar a utilização dos recursos de hardware e minimizar a perda e atraso no tráfego de pacotes [13]. Os primeiros métodos AQM utilizam basicamente o tamanho médio da fila para detectar o congestionamento, como o RED (*Random Early Detection*) [14] e ARED (*Adaptive Random Early Detection*) [17]. Os métodos AQM mais eficientes disponíveis atualmente utilizam estratégias de permissão para rajadas de pacotes, detecção do nível de atraso de permanência dos pacotes na fila, como o CoDel (*Controlled Delay Management*) [18], e mecanismos de teoria de controle para ajuste dos parâmetros, como o PIE (*Proportional Integral Controller Enhanced*) [19].

A seguir serão apresentados alguns dos métodos de AQM mais relevantes encontrados em estudos na área de pesquisa relacionada ao gerenciamento ativo de filas. São eles: RED, ARED, CoDel e PIE.

3.2.1 RED

O RED (*Random Early Detection*) [14], foi um dos primeiros métodos AQM desenvolvidos e serviu de alicerce para o desenvolvimento de novos algoritmos de gerenciamento de filas [13]. O algoritmo realiza descartes antecipados e randômicos de pacotes e a indicação de congestionamento ocorre baseada no tamanho médio da fila avg , através de uma média móvel exponencial (*Exponential Weighted Moving Average - EWMA*), possuindo dois limiares, o inferior (min_{th}) e o superior (max_{th}). Se o valor médio do tamanho da fila estiver abaixo do min_{th} , o RED não descarta pacotes. Caso esse valor seja maior que min_{th} , porém menor que max_{th} , o AQM descarta pacotes com probabilidade proporcional ao tamanho médio da fila. Quando a média da fila excede o max_{th} , todos os novos pacotes são descartados.

A cada nova chegada de um pacote à fila, os limiares min_{th} e max_{th} são comparados com a estimativa do tamanho médio da fila avg , o qual tem o seu valor atualizado dado por:

$$avg = (1 - w_q) \cdot avg + (w_q \cdot q) \quad (3.1)$$

onde o parametro w_q corresponde ao peso, geralmente ajustado para o valor 0,002, enquanto que q é o tamanho atual da fila.

O algoritmo RED divide-se em módulos de monitoramento, medição e controle do nível de ocupação das filas, obtendo a probabilidade de descarte e marcação de pacotes, garantindo, assim, a estabilidade do tamanho da fila independentemente das variações das condições da rede. Ao perceber que determinado limite de ocupação foi atingido, o RED realiza um descarte aleatório, baseado na probabilidade de descarte dada pela marcação dos pacotes.

A probabilidade de descarte p_a é calculada em função do tamanho médio da fila avg . A relação é obtida por:

$$p_b = max_p \cdot \frac{(avg - min_{th})}{(max_{th} - min_{th})} \quad (3.2)$$

onde primeiramente se calcula o valor de p_b que é utilizado como parâmetro para o cálculo final do valor de p_a , que por sua vez é dado por:

$$p_a = \frac{p_b}{(1 - count \cdot p_b)} \quad (3.3)$$

O parâmetro max_p corresponde ao máximo valor da probabilidade de descartar ou marcar pacotes na fila. A probabilidade de descarte p_a é suavemente aumentada conforme o valor da variável $count$ que corresponde ao número de pacotes que chegaram à fila desde o último descarte ou marcação e é incrementada a cada cálculo de probabilidade desde o último pacote marcado ou descartado.

O tamanho médio da fila é calculado de forma contínua, sendo utilizado como referência para determinar se haverá ou não o descarte de pacotes [14]. O RED efetua descartes de pacotes dos fluxos de maneira estatística antes mesmo de alcançar o limite de capacidade total de armazenamento da fila. O algoritmo básico do RED é apresentado em seguida:

Algoritmo 1 RED [16]

```

1: for all A cada pacote que chegar à fila; do
2:   Calcule o tamanho médio da fila  $avg$ ;
3:   if  $avg \leq min_{th}$  then
4:     Enfileire o pacote;
5:   end if
6:   if  $min_{th} < avg < max_{th}$  then
7:     Calcule a probabilidade  $p_a$ ;
8:     Descarte ou marque o pacote com probabilidade  $p_a$ ;
9:   end if
10:  if  $avg \geq max_{th}$  then
11:    Descarte ou marque o pacote;
12:  end if
13: end for

```

Ao realizar um descarte aleatório, o RED consequentemente comunica as fontes de tráfego TCP que há iminência de congestionamento na rede, fazendo com que os nós notificados diminuam sua taxa de transmissão, evitando assim o colapso na rede devido a sincronização global das retransmissões nas fontes TCP ou até a execução consecutiva de mecanismos como *Slow Start* provocando longos períodos de baixa taxa de transmissão, o que pode reduzir bastante o desempenho da rede [14]. A Figura 3.1 ilustra o comportamento das fontes de transmissão durante a ocorrências da sincronização global das fontes TCP, onde o fluxo 1 e o fluxo 2 ao perceberem o congestionamento, por meio da ausência das confirmações de recebimento dos pacotes pelo destino, baixam significativamente suas taxas de transmissão de forma orquestrada, fazendo com que o fluxo agregado subutilize a largura de banda disponível.

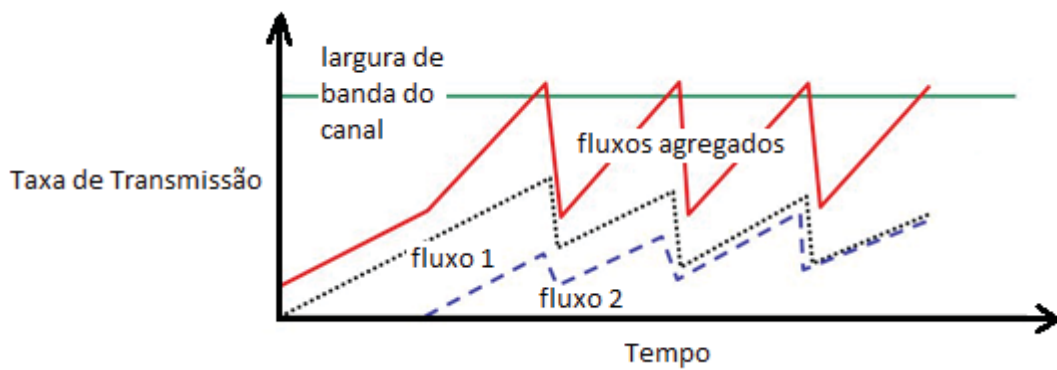


Figura 3.1: Representação do efeito da sincronização global das fontes TCP.

O RED também pode ser utilizado em conjunto com a extensão ECN do protocolo IP. O desempenho do RED está diretamente relacionado com os ajustes dos parâmetros do algoritmo, sendo bastante sensível ao tipo de tráfego envolvido e configuração da rede [17].

3.2.2 ARED

O ARED (*Adaptive Random Early Detection*) [17] consiste em uma proposta de melhoria do RED por meio de adaptações que realizam o ajuste dinâmico de parâmetros conforme condições instantâneas, incrementando a robustez do algoritmo original. Uma fraqueza bem conhecida do RED é que o *throughput* é bastante sensível à carga de tráfego e aos seus parâmetros [17].

De acordo com Floyd et al. [17], o RED não funciona bem quando o tamanho médio da fila se torna maior que max_{th} , reduzindo o rendimento e aumentando a queda de pacotes. Se o comprimento médio da fila oscila em torno do limite mínimo, a detecção precoce é muito agressiva. Por outro lado, se o comprimento médio da fila oscila em torno do limite máximo, a detecção antecipada está sendo muito conservadora.

Sendo assim, o algoritmo traz inovação ao realizar ajustes dinâmicos, alterando a probabilidade de acordo com a intensidade do tamanho médio da fila detectada, otimizando o mecanismo de descarte de pacotes. Dessa forma, elimina a sensibilidade aos parâmetros que afetam o desempenho do RED original, atingindo com maior segurança um comprimento médio de fila adequado para uma ampla variedade de cenários de tráfego.

O parâmetro max_p é adaptado lentamente, ao longo do tempo, em escalas maiores do que um tempo de RTT (*Round-Trip Time - RTT*). O RTT corresponde ao tempo total necessário para enviar um pacote da origem ao destino e obter o recebimento de confirmação de entrega na origem.

Buscando manter o tamanho médio da fila dentro de um valor alvo entre min_{th} e max_{th} , o algoritmo utiliza os fatores de incremento e decremento α e β que podem, dependendo da configuração utilizada, aumentar ou diminuir o valor de max_p conforme valor alvo estabelecido para o tamanho médio da fila.

Uma alternativa para a adaptação de max_p é descrita por Floyd et al. [17]. De acordo com os autores, o algoritmo não é uma solução ótima, mas parece funcionar bem em uma ampla gama de cenários. O algoritmo básico de adaptação dinâmica para o parâmetro max_p é descrito a seguir:

Algoritmo 2 Adaptative RED [20]

```

1: for all A cada atualização de tamanho médio da fila  $avg$  do
2:   if  $min_{th} < avg < max_{th}$  then
3:     status  $\leftarrow$  Medio
4:   end if
5:   if  $avg < min_{th} \ \&\& \ status \neq$  Abaixo then
6:     status  $\leftarrow$  Abaixo
7:      $max_p = \frac{max_p}{\alpha}$ 
8:   end if
9:   if  $avg > max_{th} \ \&\& \ status \neq$  Acima then
10:    status  $\leftarrow$  Acima
11:     $max_p = max_p \cdot \beta$ 
12:   end if
13: end for

```

3.2.3 CoDel

CoDel (*Controlled Delay Management*) [18] é um método AQM projetado para fornecer uma solução para o problema do *bufferbloat*. O método AQM CoDel realiza o controle do tempo de permanência do pacote na fila, e consequentemente limita o tamanho da desta, buscando assim manter a baixa latência na fila ao mesmo momento em que permite rajadas de tráfego, se adaptando dinamicamente a diferentes taxas de transmissão e tempos de RTTs.

O CoDel utiliza duas principais variáveis: *target* e *interval*. Em conformidade com a RFC 8289 [21], os valores ideais de *target* são 5-10% da conexão RTT e 100 ms para *interval*, de forma a satisfazerem o uso normal da Internet. Sua operação é baseada na criação de um *timestamp* do momento de chegada do pacote e ao retirar o pacote da fila é calculado o tempo de permanência do pacote, utilizando o valor de *interval* como base do mecanismo para alternar entre os estados de descarte e não-descarte.

O algoritmo inicia no estado de não-descarte e pode mudar seu estado conforme valor de tempo de permanência observado, sendo ajustado conforme o número de descartes consecutivos. O algoritmo não realiza o descarte de pacotes imediatamente quando o limite de latência é ultrapassado, pois aguarda um período de RTT para confirmar se o congestionamento é persistente, para então só depois iniciar o processo de descarte.

Se o valor de atraso observado for menor ou igual que o valor alvo em *target*, o pacote que está mais velho da fila é enviado para seu destino e o valor de *interval* é ajustado novamente para 100 ms. Se o pacote permanecer na fila mais tempo que o determinado pelo valor limite de *target*, e esta condição permanecer para todos os subsequentes pacotes pelo menos pelo valor do intervalo setado em *interval*, então o algoritmo entrará no estado de congestionamento.

No estado de congestionamento, a função de controle será acionada, excluindo um pacote e agendando tempo para o próximo evento, sendo recalculado como uma função do número de pacotes consecutivos que foram descartados, que por sua vez é inversamente proporcional a raiz quadrada do número de pacotes descartados da fila. Sendo assim, quando o algoritmo se encontra em estados sucessivos de descartes, o valor de *interval* é reduzido com a regra $interval/\sqrt{2}$, $interval/\sqrt{3}$, etc.

Segundo Nichols [18], o núcleo do problema para detecção do *bufferbloat* é saber separar os dois diferentes tipos de filas: as boas e as ruins. Em filas boas, a alta ocupação da fila desaparece em cerca de um RTT, enquanto em filas ruins persiste por vários RTTs. O CoDel busca detectar uma fila ruim, sendo uma fila que cresce de forma nociva sem demonstrar sinais de esvaziamento. Ao detectar uma fila "ruim", o algoritmo inicia uma fase de descarte

de pacotes para induzir a ativação do controle de congestionamento do TCP. Portanto o CoDel procura alcançar um modo robusto de separar os dois tipos de filas, de modo a obter o mínimo do comprimento da fila em uma janela de tempo variável que seja maior que o RTT nominal.

O CoDel é projetado para ser auto-configurável, evitando dessa forma a utilização de parâmetros que possam conduzir ao baixo desempenho. O algoritmo possui a capacidade de se adaptar às mudanças nas taxas dos *links*, além de ser projetado para uma fácil implantação e experimentação. O CoDel é bastante sensível para um valor excessivamente baixo para *interval*, já que uma queda desnecessária ocorreria e diminuiria a utilização do link [21].

3.2.4 PIE

O PIE [19] (*Proportional Integral Controller Enhanced*) é um método AQM que combina os benefícios das funcionalidades de descarte randômico realizado pelo RED com o mecanismo de detecção de congestionamento em função do atraso médio da fila proposto pelo CoDel. A probabilidade de descartar ou marcar pacotes é dinamicamente calculada com base em uma estimativa do tempo de espera na fila, ao invés do tamanho médio da fila utilizado pelo RED. O cálculo do tempo de espera na fila é uma função da estimativa de uma taxa de partida, ao invés de uma medição utilizando *timestamp* como é feito no CoDel. Há também uma previsão para chegadas de pacotes em rajadas, que em caso de estar habilitada os pacotes passam pela fila sem a aplicação da probabilidade de descarte.

O algoritmo é baseado em um controlador PI (*Proportional Integral*). O termo proporcional refere-se ao erro estimado subtraindo-se o valor corrente da variável controlada e o *set point* que se refere ao valor alvo. Porém, um controlador construído somente com o fator proporcional tende a tornar-se instável. Já um controlador que une as ações integral e proporcional são mais estáveis. O termo integral leva em consideração os valores passados do erro estimado, de forma a obter estabilidade na resposta do controlador.

Para controlar a probabilidade de descarte o método utiliza informações de atraso dos pacotes na fila e também leva em consideração a tendência de atraso, fornecendo uma sofisticada solução para determinar o atraso atual $delay_{curr}$ sem a necessidade de realizar cálculos a cada pacote recebido. O PIE também usa as tendências de aumento ou diminuição da latência para ajudar a determinar os níveis de congestionamento. Possui um mecanismo de permissão de rajadas de pacotes, evitando descartes em momentos de altas utilizações de curto prazo. Sendo assim, diferente do Codel, o PIE não requer processamento extra por pacote, pois não é necessário nenhuma inserção de *timestamp*, evitando a sobrecarga.

O algoritmo mantém uma estimativa de taxa de saída de pacotes, de forma que quando há mais pacotes na fila do que o limiar estimado em $dq_{threshold}$, um novo ciclo de medição é iniciado e termina quando o número limite de pacotes for transmitido [22]. No final do ciclo de medição, a taxa de partida é calculada dividindo a quantidade de dados transmitidos pelo tempo gasto. A Figura 3.2 apresenta, de forma simplificada, os principais componentes que compõem o método AQM PIE e suas respectivas relações.

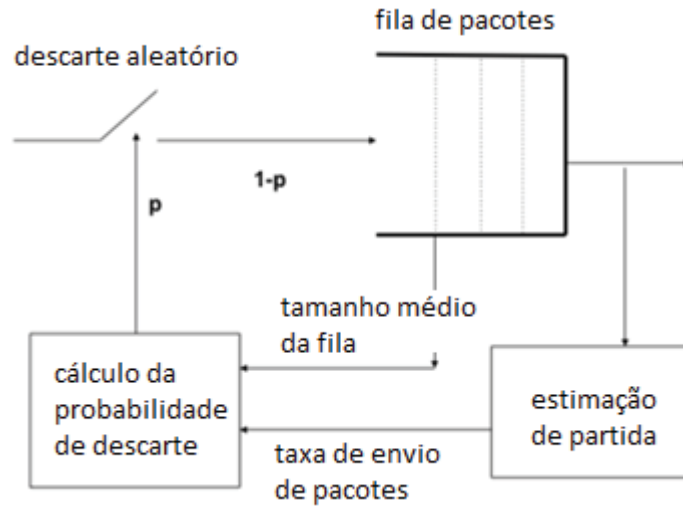


Figura 3.2: Esquema de integração entre os principais componentes utilizados pelo método AQM PIE.

A taxa média de envio é calculada usando uma média móvel do tipo EWMA (*Exponentially Weighted Moving Average*). A probabilidade de descarte p calculada é usada para os pacotes que chegam na fila. O atraso de referência $delay_{ref}$ é o atraso máximo a ser atingido. Um temporizador periódico é utilizado para atualizar a probabilidade de descarte p , obtendo o valor do atraso corrente conforme ajuste dos fatores α e β juntamente com o atraso de referência $delay_{ref}$ e atraso passado $delay_{old}$, de acordo com:

$$p = p + \alpha \cdot (delay_{curr} - delay_{ref}) + \beta \cdot (delay_{curr} - delay_{old}) \quad (3.4)$$

Os parâmetros α e β determinam, respectivamente, o quanto há de divergência do atraso de referência e quanto a tendência de atraso afeta a probabilidade de descarte [22]. O algoritmo automaticamente ajusta os parâmetros α e β com base na probabilidade de descarte calculada para dimensionar a escala de tempo de resposta. O método também implementa um mecanismo de permissão de rajadas de curto prazo no qual nenhum descarte aleatório ocorre desde que haja uma quota restante de licenças para tais rajadas. A cota é reduzida sempre que uma medição da taxa de envio for bem sucedida. Se o atraso estiver abaixo do atraso de referência, a cota de permissão de rajadas será redefinida para o valor máximo.

O mecanismo de controle do algoritmo PIE se divide em três etapas principais: o cálculo da probabilidade de marcar ou descartar pacotes, cálculo da taxa de partida de pacotes e o cálculo da tolerância às rajadas. O PIE calcula dinamicamente a probabilidade de descartar ou marcar pacotes com base em uma estimativa do tempo de espera na fila em função da estimativa de uma taxa de partida, avaliada após a saída de cada pacote. A probabilidade de descarte é atualizada periodicamente baseando-se da quantidade de pacotes desenfileirados e do limiar de atraso, além de utilizar um valor máximo permitido para que rajadas de pacotes sejam alocadas no *buffer*. O método incorpora em seu algoritmo uma previsão para chegadas em rajadas, que em caso de estar habilitada os pacotes passam pela fila sem a aplicação da probabilidade de descarte.

Assim como o CoDel, os parâmetros são de auto-ajuste. O ajuste automático de parâmetros é utilizado para manter estabilidade, sendo projetado para responder rapidamente às mudanças repentinas na rede. PIE é robusto e otimizado para vários cenários de rede, e segundo os autores, é bastante simples de implementar. Os resultados demonstrados no estudo de Pan et al. [19] garantem que o PIE é estável para um número arbitrário de fluxos com RTTs

heterogêneos, garantindo baixa latência e atingindo alta utilização de link sob várias situações de congestionamento.

3.2.5 Análise Comparativa entre os Métodos

A tabela 3.1 demonstra as características fundamentais de cada método de gerencia ativa de filas discutido anteriormente, promovendo uma maneira simples de se realizar a visualização comparativa entre os diferentes métodos AQM utilizados no contexto deste estudo.

Tabela 3.1: Comparação entre os métodos AQM discutidos.

Método	Deteção de Congestionamento	Parametrização	Permissão para Rajadas
RED	baseado no tamanho médio da fila	manual	não, sempre descarte aleatório
ARED	baseado no tamanho médio da fila	manual com ajuste dinâmico	não, sempre descarte aleatório
CoDel	baseado na latência média da fila	auto-parametrizável	sim, por meio de um <i>timer</i>
PIE	baseado em tamanho da fila e latência	auto-parametrizável	sim, até um determinado limite

3.3 Codificação de Vídeo

Um vídeo é basicamente a reprodução de uma sequência de imagens estáticas que nos fornece uma sensação de movimento entre as mesmas. A codificação de imagens envolve a representação dos vários *pixels* que a compõe em uma determinada imagem em uma série de *bits*. Quanto maior a quantidade de *pixels* de uma imagem, maior será a quantidade de *bits* necessários. Técnicas de compressão de vídeo permitem reduzir a quantidade de dados necessária para representar a imagem, o que impacta diretamente nos custos de transmissão e armazenamento de vídeos.

O dispositivo ou software conhecido por *codec* tem por objetivo realizar a compressão dos dados que compõe o vídeo original, reduzindo o seu tamanho em muitas vezes, sendo capaz de transformar representações não comprimidas de imagens em sequências menores de *bits*.

A figura 3.3 [7] apresenta uma sequência de quadros que se vistos em uma fração rápida de tempo dão a ilusão de movimento, exemplificando princípio da codificação de vídeo por meio da representação de imagens consecutivas.



Figura 3.3: Sequência de quadros que formam o vídeo *Football*.

A codificação de vídeo explora as redundâncias espaciais e temporais das imagens que compõe o vídeo. Para que a codificação de vídeo consiga obter alta eficiência na compressão, são necessários o uso de diversos recursos tecnológicos bastante utilizados nas áreas que envolvem processamento de sinais digitais, dentre os quais pode-se destacar os esquemas de predição temporal e espacial.

A predição espacial, também denominada intra-quadro, trata da redundância existente entre os elementos que constituem a cena em questão, por meio do uso de redundâncias contidas em uma mesma imagem. A transformada do cosseno [23] consiste em um método de compactação espacial bastante utilizado nos *codecs* atuais.

A predição temporal, também conhecida por predição inter-quadros, trata da redundância temporal existente entre um determinado quadro e seus antecessores e posteriores, aproveitando as informações contidas para geração de quadros de referência.

O uso de grupos de imagens, técnica conhecida por GoP (*Group of Pictures*) [23], é uma estratégia de compactação temporal, estando diretamente relacionado ao nível aplicado de predição temporal. Nos *codecs* mais difundidos é comum a existirem três tipos de quadros principais conforme descrito a seguir:

- Quadros do tipo *I* que são compactado apenas espacialmente.
- Quadros do tipo *P* que são codificados com informações baseadas no último quadro *P* ou *I*.
- Quadros do tipo *B* que são codificados utilizando informações do último quadro *P* ou *I* e do próximo quadro *P* ou *I*.

Os *codecs* geralmente possuem uma diversidade grande de parâmetros que definem a estrutura do grupo de imagem a ser utilizado na codificação de um determinado vídeo. O tamanho do GoP e sua composição, como o número de quadros do tipo *B* e *P* estão diretamente relacionados com o grau de compactação resultante e com a qualidade da imagem codificada. Cada GoP contém apenas um quadro *I*.

De forma simplificada pode-se dizer que quanto maior for o tamanho do GoP, maior será a eficiência da compactação, porém resultando numa qualidade mais baixa de imagem. De modo análogo, pode-se inferir que um GoP com tamanho reduzido, contendo poucos quadros *P* e *B*, terá menor eficiência de compactação com maior qualidade de imagem. A figura 3.4 apresenta a estrutura de um GoP (9,2), que possui 9 quadros entre dois quadros *I* consecutivos e 2 quadros *B* entre quadros *P*.

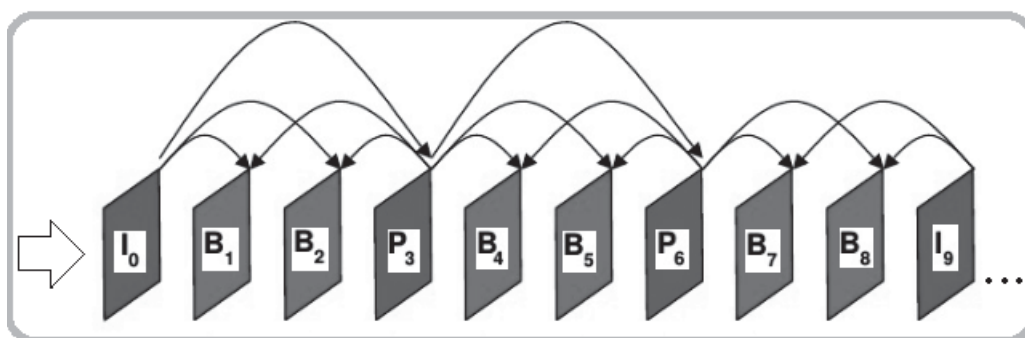


Figura 3.4: Estrutura de um GoP (9,2).

3.3.1 Advanced Video Coding - H.264/AVC

A padronização de tecnologias de codificação de vídeo visa permitir a interoperabilidade entre os equipamentos de diferentes fabricantes e os fornecedores de mídia digital. As principais organizações responsáveis pela recente padronização são o ITU (*International Telecommunication Union*) e o MPEG (*Moving Picture Experts Group*). No contexto da evolução da transmissão de conteúdo de vídeo pela Internet, tem se consolidado cada vez mais padrões de codificação de vídeo desenvolvidos pelo MPEG.

Dentre os recentes padrões largamente difundidos, destaca-se o *codec* H.264/AVC (*Advanced Video Coding*), publicado na especificação *MPEG-4 Part 10 AVC* [24]. O padrão

H.264/AVC consiste em uma avançada tecnologia de codificação de vídeo, que faz o uso de sofisticados recursos de compactação, suportando alta definição, tornando a qualidade de vídeo superior ao custo de uma taxa de bits relativamente baixa em relação aos padrões anteriores.

Através do controle flexível do fluxo de bits, o padrão torna possível a execução de vídeo de alta qualidade em diversos tipos de equipamentos, desde aparelhos de TV Digital, telefones celulares, dispositivos de TV a cabo e transmissão digital bem como tocadores de mídias físicas como o *DVDs* e *Blu-ray*, sendo adequado para qualquer plataforma da atualidade, inclusive para transmissão de vídeos pela Internet.

O aprimoramento do *codec* H.264/AVC trouxe enormes vantagens no tocante economia de espaço de armazenamento e de largura de banda para transmissão proporcionando maior qualidade de imagem, reduzindo drasticamente o volume de informações necessário à reprodução de um vídeo se comparado com as versões anteriores.

3.4 Transmissão de Vídeo pela Internet

A transmissão de vídeo de forma contínua, também conhecida por *streaming*, é uma forma de distribuição de conteúdo de vídeo por meio pacotes em uma rede, em que um usuário utiliza um *player* que obtém as informações de um servidor provedor de conteúdo, o que permite uma experiência em tempo real. Nesta forma de assistir ao conteúdo, o servidor se comunica com seus clientes através da distribuição do vídeo em fluxos contínuos, portanto não há necessidade de que as informações sejam previamente armazenadas pelo usuário em seu próprio computador para a posterior reprodução. Na Internet de hoje, as redes gerenciadas de *streaming* foram substituídos por redes de distribuição de conteúdo com implantações em larga escala de recursos intensivos pela Internet [25].

O fluxo dos dados é recebido e reproduzido à medida que chega ao usuário. O *player* consiste em um software que permite que os usuários reproduzam os arquivos multimídia. Os provedores de conteúdo distribuem os conteúdos de vídeo para os usuários, utilizando um protocolo definido, por meio de seus servidores. O *YouTube* e o *Netflix* são os principais exemplos de serviços provedores de conteúdo de *streaming* de vídeo, utilizados por usuários de muitos países. Geralmente, as informações de mídias transmitidos para o usuário ficam armazenados em servidores que possuem uma enorme capacidade de armazenamento de dados e estão conectados à Internet por meio de enlaces de alta velocidade, que permitem a transmissão de arquivos com qualidade mesmo para locais geograficamente distantes dos provedores.

Os primeiros sistemas de *streaming* de vídeo geralmente usavam protocolos com estados, como por exemplo, o RTP (*Real-Time Protocol*) e o RTSP (*Real-Time Streaming Protocol*) [26]. Nessa configuração, quando um cliente se conecta ao servidor, este controla o estado do cliente até que o cliente se desconecte novamente. Normalmente, ocorre comunicação frequente entre o cliente e o servidor depois de estabelecida uma sessão, em que o servidor envia a mídia como um fluxo contínuo de pacotes sobre o protocolo de transporte UDP [26]. O RTP e o RTSP funcionam bem em redes IP gerenciadas [25]. Porém na Internet o protocolo HTTP traz uma maior flexibilidade e compatibilidade com as demais tecnologias atuais.

Modernos sistemas de *streaming* de vídeo transmitem o conteúdo através do protocolo HTTP, que trata-se de um protocolo sem estados. Sendo assim, um cliente HTTP pode solicitar alguma porção de dados, e o servidor responderá enviando os dados e a transação será finalizada, não sendo necessário estabelecimento de uma sessão. Cada solicitação HTTP é tratada como uma transação única completamente independente e com isso há o benefício e a flexibilidade para prover o benefício da adaptabilidade dinâmica do conteúdo de vídeo por meio de pequenos segmentos transmitidos de forma contínua entre servidor e cliente [9].

3.4.1 *Dynamic Adaptive Streaming over HTTP - DASH*

Atualmente, a entrega de fluxo de vídeo de modo dinâmico e adaptativo através do protocolo HTTP (*Hypertext Transfer Protocol*) denominada DASH (*Dynamic Adaptive Streaming over HTTP*), vem se tornando um padrão para a transmissão de vídeo sobre demanda (*Video on Demand - VoD*) sendo também bastante utilizado para transmissões ao vivo (*Live streaming*) [27]. Com o uso do padrão DASH, os vídeos são codificados em diferentes qualidades e temporalmente divididos em segmentos de alguns poucos segundos, constituindo representações apropriadas para determinadas taxas de transferência. Sendo assim, após o dispositivo cliente iniciar a sessão com o servidor, de acordo com mecanismos de estimação das condições da rede, realiza a requisição dos segmentos em determinada qualidade, podendo alternar entre as representações disponíveis.

De acordo com [28], de modo heurístico, o cliente decide em qual qualidade cada segmento deve ser solicitado, baseando-se em parâmetros como taxa de transferência e o atraso. Isso possibilita que o DASH se adapte às flutuações que ocorrem na largura de banda, permitindo que o cliente busque segmentos de menor qualidade, mantendo a execução do vídeo constante mesmo em cenários de congestionamentos [29].

O grupo de trabalho responsável pela especificação do MPEG-DASH é composto de uma grande parcela da indústria especializada, incluindo partes interessadas e contribuidores como as empresas *Apple*, *Adobe*, *Microsoft*, *Netflix*, dentre outros [25]. As plataformas modernas de distribuição de vídeo pela rede mundial de computadores, adotaram o DASH como a principal técnica de entrega de vídeo aos clientes [30]. A fim de propor um padrão para a entrega de fluxos de vídeo sobre o HTTP *Hypertext Transfer Protocol*, buscando compatibilidade entre os distintos fornecedores, o MPEG padronizou a solução denominada MPEG-DASH [27].

Embora existam divergências de implementações entre as soluções existentes, todas se baseiam nos seguintes princípios: no servidor, o vídeo é codificado em diferentes taxas de qualidade e dividido em segmentos. O cliente dinamicamente busca os segmentos que melhor se adaptem a atual largura de banda, baseado em métricas como taxa de transmissão, atraso, *jitter* e estado do *buffer* de reprodução. A figura 3.5 apresenta o mesmo quadro do vídeo *Sunflower* [7] em 2 diferentes representações de qualidade, sendo o da esquerda com maior qualidade, contendo maior taxa de *bits*, e o da direita com menor qualidade, com menor taxa de *bits*.

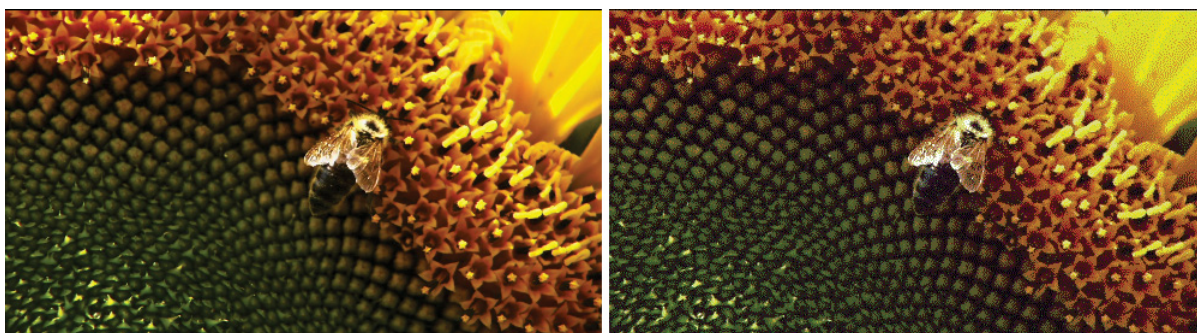


Figura 3.5: Quadro do vídeo *Sunflower* em 2 diferentes representações de qualidade.

Ao iniciar uma sessão, o cliente envia uma requisição ao servidor, solicitando o arquivo de manifesto conhecido por MPD (*Media Presentation Description*). Nele, uma série de itens são descritos como o tipo do vídeo (ao vivo ou sobre demanda), tempo de duração total do vídeo e de um segmento, qualidades disponíveis, valores da largura de banda de cada taxa de representação, entre outros.

Após receber e analisar o arquivo MPD, o cliente passa a fazer requisições pelos segmentos do vídeo, como apresenta a Figura 3.6. A cada nova requisição ou a partir de determinado critério para ajuste de qualidade, o dispositivo cliente pode realizar a estimação de parâmetros, como a largura de banda disponível, utilizando-se de diversas variáveis, como o tempo que o segmento levou para ser entregue, seu tamanho em *bytes* ou até mesmo a sua duração em tempo de reprodução. Assim, o próximo trecho de vídeo a ser requisitado deve se adequar ao melhor resultado obtido pela estimação dos parâmetros, como por exemplo, ter a largura de banda próximo ou abaixo do valor de determinado instante [31]. A Figura 3.6 ilustra o comportamento adaptativo do DASH ao longo do tempo.

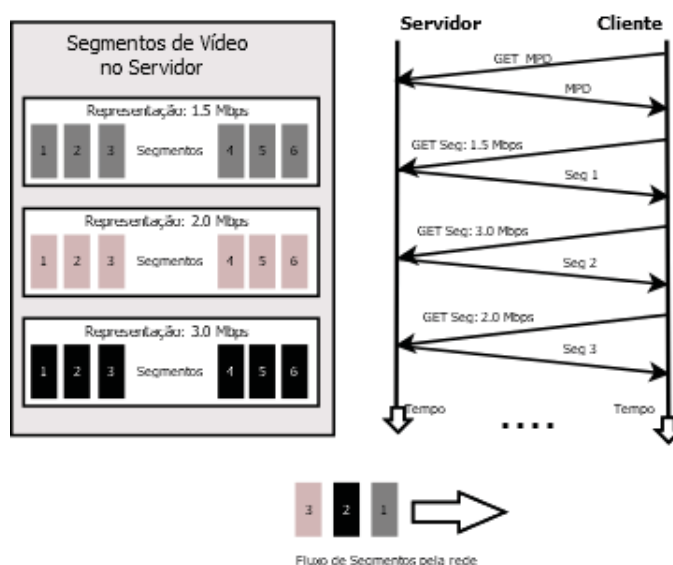


Figura 3.6: Comportamento adaptativo do DASH ao longo do tempo.

O padrão DASH independente do *codec* utilizado, portanto preconiza um formato aberto de distribuição de vídeo pela Internet, proporcionando interoperabilidade entre diferentes desenvolvedores de softwares clientes e servidores de *streaming*, impulsionando assim um ambiente comum de conteúdos e serviços de vídeo online, além de buscar compatibilidade com os diversos tipos de dispositivos e suas respectivas capacidades de reprodução.

Por ser baseado no protocolo HTTP, o DASH possui a vantagem dos pacotes atravessarem facilmente dispositivos como *Firewalls*, IPS (*Intrusion Prevention System*) e sistemas que implementam o NAT (*Network Address Translation*), sem a necessidade de configurações específicas para protocolos distintos. Adicionalmente, a transmissão de vídeo por *HTTP* possibilita o uso de mecanismos de *Cache* distribuídos pela *Web*, uma vez que os dados podem estar localizados geograficamente próximos do visualizador, tornando mais rápida e eficiente, transmissão e reprodução de segmentos de vídeo, melhorando assim a qualidade do serviço oferecido [25].

3.4.2 DASH e o Estado da Arte

Há diversos estudos que analisam casos particulares relativos à transmissão de fluxo de vídeo adaptativo utilizando tecnologia DASH, como o trabalho de Le Feuvre et al. [32] abrange um novo mecanismo de estimação para seleção de qualidade de segmentos de vídeo que analisa a latência ideal para os canais de transmissão. Já o estudo de Concolato et al. [33] apresenta as vantagens da utilização de *HEVC Tiled Videos* que permite um encapsulamento baseado em

blocos de entrega. Por sua vez, Lohmar et al. [34] realiza uma análise aprofundada de DASH com o uso de conteúdo de vídeo ao vivo (*live content*) identificando os diferentes componentes de atraso e como podem contribuir para minimizar os efeitos do atraso de ponta a ponta dos serviços em tempo real.

Trabalhos como de Lederer et al. [35], Le Feuvre et al. [36], Kreuzberger et al. [37] e Quinlan et al. [38], focam na disponibilização de ferramentas e repositórios de segmentos de vídeos (*datasets*) para testes diversos de transmissão de fluxo adaptável de vídeo com o padrão DASH, podendo ser utilizados pela comunidade de pesquisa para avaliar o desempenho de softwares compatíveis bem como analisar o desempenho na reprodução de conteúdo de vídeo codificado em diversas representações.

Já os estudo de Kua et al. [31] fornece uma visão profunda das diversas técnicas existentes para realizar a estimação das condições instantâneas da rede para fluxos DASH, realizando a avaliação de diversos parâmetros como RTT, *buffer* e variação do atraso, necessários para seleção da qualidade ideal para determinado momento, obtendo a melhor alternativa para que se efetive a adaptação dinâmica de conteúdo na utilização de fluxos adaptativos.

3.5 Aplicações de Métodos AQM

Segundo Adams [13] há um grande empenho em áreas de pesquisa correlatas à novos modelos e avaliação de performance de algoritmos para gerenciamento ativo de filas. De acordo com Patil e Tahiliani [39], a utilização de técnicas de gerenciamento ativo de filas possibilitam a melhora do tráfego de pacotes até mesmo quando as fontes de tráfego envolvidas não colaboram no controle de congestionamento da rede como o protocolo UDP (*User Datagram Protocol*).

O estudo de Kim et al. [40], demonstra que a utilização de mecanismos de AQM na camada de rede auxiliam na diminuição da taxa de perda de pacotes e aumentam a qualidade de vídeo percebida pelos usuários na utilização de serviços multimídia, principalmente em transmissões através de enlaces congestionados.

A RFC 8290 [41] apresenta o método FQ-CoDel (*Flow Queue-Controlled Delay*) que adiciona ao método AQM CoDel os benefícios de um mecanismo de escalonamento de pacotes. O algoritmo classifica os pacotes que chegam em uma das filas FIFO disponíveis, podendo selecioná-los de acordo com determinados critérios, como endereços IP e portas de origem e destino. Um escalonador DRR (*Deficit Round Robin*) [42] modificado, retira os pacotes das filas, dando prioridade aquelas que contém fluxos recém chegados. Cada fila é separadamente atendida por um mecanismo CoDel. O enfileiramento baseado no escalonador DRR é uma técnica que permite que cada fluxo que passa por um dispositivo de rede tenha uma participação justa nos recursos da rede e alcance uma igualdade quase perfeita em termos de taxa de transferência, exigindo pouca sobrecarga para processar um pacote. Sendo assim, com o FQ-CoDel, cada fila é servida separadamente por um método AQM CoDel.

3.5.1 Melhoria da qualidade de *Video Streaming* com AQM

Kua e Armitage [30] avaliam os efeitos da aplicação dos métodos AQM PIE e FQ-CoDel em transmissões de vídeo com a tecnologia DASH, além de apresentar e avaliar um novo método denominado FQ-PIE, um esquema híbrido que envolve os benefícios do PIE e FQ-CoDel. Os resultados indicam que o PIE possui maior tolerância às rajadas enquanto que o FQ-PIE fornece maior proteção aos fluxos DASH na presença de tráfego, cruzado proporcionando melhor desempenho.

Uma análise sobre os benefícios da implementação de métodos AQM como o PIE e o FQ-CoDel para fluxos de vídeo DASH também é apresentada por Kua e Armitage [2]. Os resultados demonstram que o isolamento de fluxo e a capacidade de justiça no compartilhamento de tráfego do FQ-CoDel proporciona uma melhor experiência do usuário na presença de fluxos elásticos competitivos. Os resultados mostram melhores desempenhos do PIE e FQ-CoDel quando comparado ao DropTail, em termos das transições das taxas de qualidade do vídeo bem como requisições de segmentos de melhores qualidade em cenários de congestionamentos.

A fim de oferecer fluxo de vídeo em tempo real, de modo confiável e sem interrupções ao cliente, Bouten et al [27] propõe um método de gerenciamento de fila que prioriza a transmissão dos segmentos de menor taxa de qualidade, em cenários de congestionamentos na rede. De acordo com os autores, os sistemas DASH tratam igualmente clientes que estão na eminência de terem suas memórias vazias, pela falta da chegada de novos segmentos, daqueles que possuem segmentos suficiente para atravessarem períodos de congestionamento. Os autores apresentam que o método melhora o MOS (*Mean Opinion Score*) [43] e a quantidade de transições entre os segmentos quando comparado à transmissão DASH sem prioridades.

O estudo de D.P. Torres et al. [44] investiga o impacto na QoS oferecida pelas redes IP que transportam fluxos de vídeo H.264/MPEG-4/AVC em tempo real ao aplicar o método de gerência ativa de filas ARED em comparação ao método de descarte tradicional DropTail. Os resultados das simulações indicam que a técnica AQM ARED reduz a perda de pacotes de vídeo, mas degrada um número maior de quadros de vídeo se comparado à técnica DropTail, enquanto promove um pequeno ganho qualitativo em relação a atraso e *jitter*.

Sistemas de transmissão de vídeo no padrão DASH utilizam mecanismos de adaptação que envolvem principalmente a camada de aplicação, aspecto ainda não muito explorado pelos métodos para gerência ativa de filas. O desenvolvimento de algoritmos de AQM tem sido fundamental para suportar a transmissão de vídeo com qualidade, e permanece ainda como um tópico aberto ao desenvolvimento de pesquisa, principalmente considerando-se os recentes desenvolvimentos na tecnologia de compactação de vídeo e adaptação de conteúdo.

Capítulo 4

Método Proposto

4.1 Paradigma do Modelo

O método proposto foi projetado para transmissões de fluxo de vídeo adaptativo ao vivo que requer latência mínima para perfeita reprodução no cliente. Realiza o descarte antecipado aleatório randômico baseado no tempo de permanência do pacote na fila do roteador utilizando os limiares baseados no tempo de permanência do pacote na fila do roteador t_{min} e t_{max} . O método se fundamenta no tempo máximo de latência de um pacote de vídeo para utilidade na reprodução no destino.

A proposta de mecanismo gerenciamento ativo de filas se baseia na estimação do tempo de permanência dos pacotes na fila, descartando pacotes de forma antecipada ao enchimento da fila, semelhante ao método AQM RED que considera a ocupação relativa da fila para descarte antecipado.

O princípio de funcionamento do algoritmo visa estabelecer o tempo máximo de chegada de um pacote de vídeo para sua perfeita reprodução no destino. Se o tempo de permanência estimado de um pacote na fila for maior do que o limite para o mesmo chegue no momento de sua reprodução no cliente, significa que não há mais sentido na entrega desse pacote ao destino, portanto o mesmo deve ter maior probabilidade de ser descartado.

O descarte antecipado tem o objetivo de fazer com que as fontes de tráfego reduzam a taxa de transmissão contribuindo para a redução na latência antes que grandes rajadas de tráfego sejam perdidas, resultando em melhor qualidade média percebida pelo usuário na reprodução do vídeo. Sendo assim, o modelo proposto possui foco em aplicações de *streaming* de vídeo adaptativo, sendo projetado para transmissões de vídeo ao vivo como telejornais e eventos esportivos, com aplicação principalmente em enlaces com nível médio e alto de congestionamento.

A política enfileiramento é baseada no FIFO (*First In First Out*) com o descarte aleatório antecipado utilizando o tempo estimado de permanência do pacote na fila como parâmetro para intensificar ou reduzir a probabilidade de descarte dos pacotes que se encontram enfileirados.

4.2 Parâmetros e Algoritmo

Os principais parâmetros utilizados no modelo AQM proposto são:

- t_i : tempo estimado, em mili-segundos (ms), de permanência do pacote x_i recebido;
- t_{min} : tempo mínimo, em mili-segundos (ms), para permanência do pacote na fila;
- t_{max} : tempo máximo, em mili-segundos (ms), para permanência do pacote na fila;

- p_i : probabilidade de descarte para o pacote x_i , variando entre 0.1% e 99.9%.

O tempo estimado para permanência de um pacote x_i na fila é dado por:

$$t_i = \sum_{j=1}^i \frac{b_j}{C} \quad (4.1)$$

onde b_j é o tamanho do pacote em bits e C é a capacidade do enlace em bits por segundo.

A cada pacote x_i recebido será calculado o valor de t_i . O cálculo de t_i leva em consideração todos os i pacotes que se encontram na fila naquele determinado instante. Caso t_i seja menor que o limiar de tempo mínimo t_{min} , o pacote x_i será enfileirado. Caso t_i exceda o limiar inferior t_{min} , porém ainda seja menor que o limiar superior t_{max} , o algoritmo realiza descartes aleatórios conforme probabilidade p_i . Caso t_i exceda o limiar de tempo máximo t_{max} , o pacote x_i será descartado ou marcado com probabilidade $p_i = 1$.

A probabilidade de descarte é diretamente proporcional ao tempo setado no limiar superior. Se o tempo de permanência t_i for muito próximo à t_{max} , a probabilidade de descarte será alta, caso contrário se o valor de t_i for próximo à t_{min} , a probabilidade de descarte será baixa.

O cálculo da probabilidade de descarte p_i é dado por:

$$p_i = \frac{t_i}{t_{max}} \quad (4.2)$$

onde t_i representa do valor de permanência para o pacote x_i e t_{max} representa o valor do limiar superior de tempo de permanência do pacote na fila.

O mecanismo de realização de descarte aleatório randômico é implementado por meio de alterações no algoritmo RED, de forma que seus limiares min_{th} e max_{th} referentes ao tamanho médio da fila passam a ser baseados nos limites temporais de permanência do pacote na fila do roteador, t_{min} e t_{max} . O método proposto foi implementado no simulador NS-3 por meio de alterações no modelo implementado do RED na classe *Traffic Control Layer RED queue disc* [45], realizando uma conversão direta entre o número de pacotes na fila e tempo de permanência dos mesmos, por meio da Lei de Little [46].

Desenvolvida por John Little no início dos anos 60, a Lei de Little relaciona o número de clientes no sistema com o tempo médio despendido no sistema. Tal relação se aplica a todos os sistemas de fila, independentemente do processo de chegada e atendimento [46].

A Lei de Little pode ser definida pelas seguintes relações:

$$NF = TF \cdot \lambda \quad (4.3)$$

$$NS = TS \cdot \lambda \quad (4.4)$$

onde NF representa o número médio de elementos na fila, NS representa o número médio de elementos no sistema, TF representa o tempo médio gasto na fila, TS representa o tempo médio gasto no sistema, e λ representa o tempo médio de atendimento.

A seguir é apresentado, de forma simplificada, o algoritmo do método proposto em pseudocódigo:

Algoritmo 3 Método Proposto

```

1: for all A cada pacote que chegar à fila do
2:    $t_i \leftarrow \sum_{j=1}^i \frac{b_j}{C}$  // Calcule o tempo de permanência do pacote  $x_i$ 
3:   if  $t_i < t_{\min}$  then
4:      $\text{queue}(x_i)$  // Enfileire o pacote  $x_i$ 
5:   end if
6:   if  $t_{\min} \leq t_i \leq t_{\max}$  then
7:      $p_i \leftarrow \frac{t_i}{t_{\max}}$  // Calcule a probabilidade de descarte  $p_i$ 
8:      $\text{random} \leftarrow \text{getRandom}()$  // Calcule o número aleatório
9:     if  $p_i > \text{random}$  then
10:       $\text{drop}(x_i)$  // Descarte o pacote  $x_i$ 
11:    end if
12:    if  $p_i \leq \text{random}$  then
13:       $\text{queue}(x_i)$  // Enfileire o pacote  $x_i$ 
14:    end if
15:  end if
16:  if  $t_i > t_{\max}$  then
17:     $\text{drop}(x_i)$  // Descarte o pacote  $x_i$ 
18:  end if
19: end for

```

Nesse modelo, um controlador SDN (*Software Defined Network*) [47] será responsável por informar ao plano de dados da rede pertencente ao domínio do provedor, as características de atraso máximo de propagação, que serão utilizados para estabelecer os limiares máximo e mínimo de tempo de permanência na fila. Para tanto, os valores de t_{\min} e t_{\max} são setados dinamicamente pelo controlador SDN conforme condições da rede, que podem ser mensuradas periodicamente através da observação de diversos parâmetros, como latência máxima permitida em função do comportamento dos fluxos envolvidos na rede de acesso. O modelo é restrito à redes SDN de domínio único.

O modelo considera que a diferença de atraso na última milha entre os diferentes clientes envolvidos é insignificante, utilizando uma estimativa média do que seria a máxima latência suportada em transmissões de vídeo ao vivo. Essa latência é distribuída uniformemente entre os roteadores que compõe a infraestrutura de acesso. Sendo assim, o controlador SDN pode realizar a alteração dos limiares para uma melhor adequação do desempenho do método AQM, de acordo com a alteração de comportamento dos fluxos que atravessam os ativos da infraestrutura de rede da acesso da operadora em um determinado momento.

4.3 Benefícios Esperados

Ao descartar pacotes preventivamente espera-se que o mecanismo de controle de congestionamento do TCP perceba e reduza a taxa de transmissão. A redução da taxa de transmissão pelo TCP consequentemente induz ao DASH selecionar os próximos segmentos de vídeo com menor qualidade. Com isso, o mecanismo de adaptação do vídeo alcança um comportamento semelhante ao que ocorre entre o TCP com o RED original, quando o TCP reduz sua taxa na iminência de um congestionamento.

Porém, neste caso também haverá colaboração com a camada de aplicação, conforme mecanismos utilizados pela tecnologia DASH, por meio da transição de qualidade entre os segmentos do vídeo. As ocorrências de descartes antecipados realizados pelo método AQM fará

com que o TCP reduza sua taxa de transmissão, o que fará com que o mecanismo de estimação do DASH detecte uma menor largura de banda disponível ou um RTT maior entre a origem e o destino.

Conforme mensuração instantânea das condições da rede, o vídeo pode se adaptar para uma qualidade menor antes que a rede entre em colapso, agindo de forma preventiva ao congestionamento, reduzindo assim a taxa de perda de pacotes e consequentemente a latência na transmissão do vídeo. Dessa forma, espera-se melhorar a qualidade média percebida pelos usuários, bem como promover melhor justiça entre os fluxos que compartilham o mesmo enlace e largura de banda.

Adicionalmente, espera-se obter maior estabilidade de qualidade durante a seleção dos segmentos de vídeo, mantendo a reprodução do conteúdo sem constantes alterações abruptamente nas representações disponíveis, o que pode não ser agradável para o expectador. A Figura 4.1 apresenta a seleção dos segmentos na linha do tempo em função das taxas disponíveis durante a transmissão do vídeo *Big Buck Bunny*, na qual os segmentos são repentinamente alterados de qualidade ao longo do *streaming*. A ideia central da proposta é reduzir a grande variação de qualidade apresentada, pois reduzindo a variação espera-se um melhor PSNR médio percebido pelo usuário.

O uso do método proposto visa garantir que o vídeo se mantenha por mais tempo em uma determinada qualidade adequada para o momento do congestionamento, proporcionando transições mais suaves e gradativas entre os segmentos, possibilitando assim uma experiência mais agradável ao usuário que assiste à transmissão do vídeo.

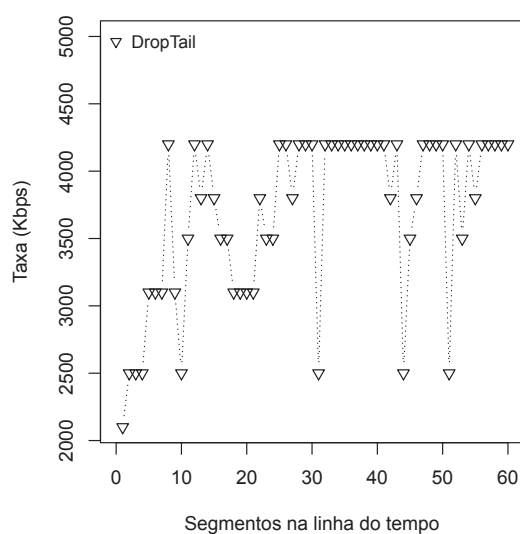


Figura 4.1: Transição entre segmentos durante a transmissão do vídeo *Big Buck Bunny* com o DropTail.

Capítulo 5

Avaliação de Desempenho

5.1 Cenário de Avaliação

A avaliação de desempenho do método AQM proposto foi realizada através da implementação de um servidor e cliente DASH real gerando tráfego em tempo real para uma topologia de rede simulada. Isto foi necessário porque os simuladores não disponibilizam módulos DASH devido a sua grande complexidade da adaptação do tráfego e rapidez na evolução dos padrões. Para tanto, foi desenvolvido um cenário de rede IP simulada, em que o servidor real transfere o fluxo de vídeo adaptativo para o cliente real concorrendo com outros fluxos pelos recursos de largura de banda disponível no enlace principal, possibilitando assim reproduzir os efeitos do congestionamento em uma rede com várias fontes distintas transmitindo simultaneamente.

Foram realizadas diversas atividades compreendendo a geração de conteúdo de vídeo a ser transmitido, o desenvolvimento do ambiente computacional com suas interfaces de interconexão ao simulador e a implementação de *scripts* para as simulações. Algumas das principais etapas executadas são descritas a seguir:

- Pesquisa e *download* de arquivos de vídeos em formato não comprimido, tradicionalmente utilizados em pesquisa na área relacionada com a qualidade de vídeo;
- Codificação dos arquivos de vídeo em diversas qualidades, gerando posteriormente os segmentos de vídeo, agrupando os mesmos nas representações correspondentes, descrevendo o conteúdo em arquivo de manifesto, gerando assim o conteúdo compatível com o padrão MPEG-DASH;
- Configuração do ambiente computacional em que as simulações foram executadas, promovendo a interconexão entre as fontes reais de tráfego reais com a rede simulada;
- Implementação no simulador do cenário de rede IP com a topologia adequada, fontes de tráfego de fundo, roteadores e algoritmos AQM;
- Transmissão do fluxo de vídeo adaptativo em diferentes níveis de utilização do enlace principal, alternando o algoritmo AQM na fila do roteador principal e coletando resultados;
- Análise de qualidade do vídeo recebido no cliente através da ferramenta PSNR, além das estatísticas de tráfego para cada experimento e comparação de resultados obtidos por cada um dos métodos de AQM, por meio do tratamento dos dados coletados e da geração de gráficos.

Para concepção da estrutura de validação do método e execução dos experimentos, foram utilizados os recursos apresentados na Tabela 5.1.

Tabela 5.1: Principais Recursos Utilizados para Avaliação de Desempenho

Recurso	Descrição
HARDWARE	Computador com sistema operacional Linux com conexão à Internet para pesquisas, análises e testes de ferramentas diversas para <i>streaming</i> de vídeo, além de implementar e realizar simulações. No presente trabalho foi utilizado um computador com sistema operacional Linux com a seguinte configuração: Processador Quadcore 3 GHz, 8GB de memória RAM, interfaces de rede GigaEthernet, suporte a virtualização total de hardware e sistemas operacionais;
SOFTWARE	Simulador de rede NS-3 para implementação do cenário de <i>streaming</i> de vídeo em rede IP e métodos AQM nas filas dos roteadores, software <i>R Project for Statistical Computing</i> para realizar o tratamento estatístico dos dados resultantes da simulação; <i>Frameworks</i> multimídia utilizados na preparação, manipulação, tratamento, conversão, codificação, testes e reprodução de arquivos de vídeo, como <i>FFmpeg</i> , <i>Vooya</i> e <i>GPAC</i> , além de software específico para disponibilização de conteúdo em servidor HTTP, como <i>Apache</i> ; software de virtualização <i>VirtualBox</i> , ferramentas para cálculo de PSNR como o <i>Evalvid</i> [48], dentre outros aplicativos necessários.
DADOS	Banco de dados de vídeos não comprimidos, codificados em formato YUV disponíveis em sites de universidades e grupos de pesquisas como o <i>YUV Video Sequences - Arizona State University</i> [7], além de repositórios para testes de ferramentas e <i>players</i> para reproduzir vídeos no padrão DASH, contendo vídeos codificados em H.264/AVC e distribuídos em sequências de diversas qualidades.

5.1.1 Conteúdo de Vídeo

Para geração de conteúdo de vídeo a ser transmitido nos experimentos, diferentes vídeos foram selecionados para evidenciar o aspecto geral dos testes, buscando proporcionar ampla diversidade de conteúdo na avaliação de desempenho. Sendo assim, foram utilizados os vídeos *Big Buck Bunny*, *Rush Hour*, *Sunflower*, *Touchdown Pass*, *Pedestrian Area* e *Riverbed*, disponíveis publicamente no *website Derf's Test Media Collection* [7].

O vídeo *Big Buck Bunny* é um desenho animado, com várias transições de cenas contendo personagens em meio a natureza. *Rush Hour* consiste em cenas de intenso tráfego de veículos em meio a ondas de calor, capturadas por uma câmera estática. *Sunflower* apresenta cenas reais em que uma câmera, em movimentos, lentos foca em uma abelha no primeiro plano e uma flor ao fundo, contendo poucas transições de cenas. *Touchdown Pass* trata-se de um vídeo que apresenta um jogo de futebol americano com transições rápidas entre cenas em que jogadores correm atrás da bola, sendo capturadas por uma câmera em movimento. *Pedestrian Area* apresenta cenas de intenso tráfego de pedestres em meio a um centro comercial, capturado por uma câmera estática. *Riverbed*, também utiliza uma câmera estática, a qual captura cenas de um leito de um rio, mostrando o constante movimento das águas, além de uma folha flutua pelas águas. A Figura 5.1 [7] ilustra um quadro intermediário de cada um dos vídeos utilizados na avaliação de desempenho.

Os vídeos originais possuem formato não comprimido, utilizando o esquema para subamostragem de cores (*color subsampling*) baseado na técnica conhecida por *Chroma subsampling*,

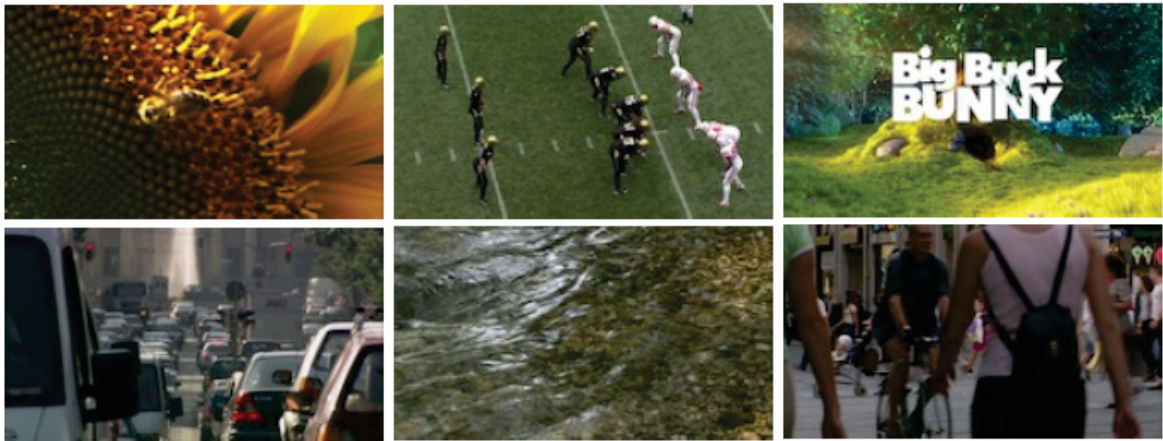


Figura 5.1: Os vídeos, em sentido horário, *Sunflower*, *Touchdown Pass*, *Big Buck Bunny*, *Pedestrian Area*, *Riverbed* e *Rush Hour*.

de YUV420 [23] para os vídeos *Big Buck Bunny*, *Rush Hour*, *Sunflower*, *Pedestrian Area* e *Riverbed* e de YUV422 [23] para o vídeo *Touchdown Pass*.

O YUV consiste em um sistema de codificação que trabalha com componentes separados de luz e cor, possuindo a representação em três componentes, sendo um para luminosidade e outros dois para informação de cor. O processo de *sampling* analisa o sinal analógico proveniente da leitura dos *pixels* que compõe a imagem e o digitaliza.

O *chroma subsampling* consiste em um esquema de codificação de imagens e vídeos através da implementação de componentes de croma e luminância, no qual utiliza mais informações de cores do que de luminância, aproveitando-se assim do sistema de visão humana que tem menor capacidade para diferenciar cores do que luminosidades, promovendo um esquema de codificação que faz com que imagens e vídeos tenham um tamanho menor em bits, reduzindo o tamanho do arquivo [23].

A Tabela 5.2 apresenta os parâmetros relativos às características técnicas dos vídeos utilizados nos experimentos, como a duração total de tempo em segundos, a quantidade total de quadros que compõe a sequência, a respectiva taxa de amostragem em quadros por segundo, além do formato original em que os vídeos foram disponibilizados para os testes.

Tabela 5.2: Parâmetros dos vídeos utilizados na avaliação de desempenho.

Vídeo	Duração	Total de Quadros	Quadros por Segundo	Formato Original
<i>Big Buck Bunny</i>	60s	1440	24	YUV 420
<i>Rush Hour</i>	20s	500	25	YUV 420
<i>Sunflower</i>	20s	500	25	YUV 420
<i>Pedestrian Area</i>	15s	375	25	YUV 420
<i>Riverbed</i>	10s	250	25	YUV 420
<i>Touchdown Pass</i>	19s	570	30	YUV 422

É importante ressaltar que cada vídeo utilizado possui diferentes características de cenas e padrão de movimento, o que permite que a avaliação de desempenho alcance maior diversidade de conteúdo nos experimentos. Sendo assim, foram levadas em consideração as medidas objetivas de padrão de movimento e de detalhes da imagem, conhecidas respectivamente por *Motion Activity* [49] e *Detail* [50].

O *Detail* é baseado no resumo dos descritores de histograma nas imagens que compõe a sequência de vídeo, enquanto o *Motion Activity* informa o grau de intensidade de movimento relacionado às alterações entre as cenas [51]. O estudo de Zakerinasab e Wang [51] classificam o grau de intensidade de movimento em um número inteiro no intervalo de 1 a 5 com o seguinte significado: 1) intensidade muito baixa; 2) baixa intensidade; 3) intensidade média; 4) alta intensidade; e 5) intensidade muito alta.

A tabela 5.3 apresenta a relação dos vídeos e suas características técnicas em relação aos valores para o *Detail* e o *Motion Activity*, além de especificar o gênero (*Genre*) conforme classificação realizada no estudo de Zakerinasab e Wang [51]. Porém, o estudo mencionado anteriormente não utilizou vídeo de esportes. Portanto, para ampliar ainda mais a diversidade de gêneros, foi acrescentado o vídeo *Touchdown Pass* nos experimentos, pois se trata de sequências de imagens de um evento esportivo, sendo muito importante a validação desse tipo de conteúdo, uma vez que diversos usuários utilizam serviços de *live streaming* para acompanhar jogos transmitidos ao vivo.

Tabela 5.3: Características técnicas de imagem e movimento dos vídeos.

Vídeo	Detail	Motion Activity	Genre
<i>Big Buck Bunny</i>	3.52	1.63	Animation
<i>Rush Hour</i>	3.17	3.12	Scene
<i>Sunflower</i>	4.04	2.57	Nature
<i>Pedestrian Area</i>	3.15	4.42	Scene
<i>Riverbed</i>	4.72	4.13	Nature

Após a seleção do conteúdo de vídeo é necessário a codificação dos mesmos em diferentes qualidades (*bitrate*). A codificação dos vídeos foi realizada de forma *offline* através da ferramenta FFmpeg [52]. O *codec* H.264/AVC foi selecionado por meio da biblioteca *libx264*. Buscando uma compactação razoável, sem que a qualidade fosse prejudicada por grandes perdas de informação, foi utilizado o GOP (*Group of Picture*) de 6 quadros, com 2 quadros *B* intercalados entre o quadro *I* e o *P* [53].

Os vídeo foram codificados no padrão H.264/AVC em seis distintas representações a seguir: 2.1 Mbps, 2.5 Mbps, 3.1 Mbps, 3.5 Mbps, 3.8 Mbps e 4.2 Mbps. Tais taxas (*bitrates*) e características de quantização são compatíveis com outros estudos correlatos [30]. Todos os vídeos utilizados possuem resolução HD (*High Definition*), contendo 1920×1080 linhas, pois trata-se do padrão mais utilizado nas atuais aplicações de *streaming* disponíveis na Internet, como se pode perceber nos principais provedores de conteúdo como *Youtube* e *Netflix*.

A geração de segmentos foi realizada de forma ao vivo, através da ferramenta *MP4Box* [54] seguindo o *profile Live-H.264* conforme especificação do padrão MPEG-DASH [55] para *live streaming*. Os segmentos de vídeo ao vivo foram gerados com 1 segundo de duração de acordo com [28]. Na configuração ao vivo, os segmentos expiram no servidor após determinado tempo de geração, o que obriga o cliente a requisitar sempre o próximo segmento disponível, promovendo assim o impacto negativo com a perda dos segmentos expirados. Se o cliente requisitar um segmento que já tiver expirado, é entregue o próximo segmento e assim por diante.

Após a codificação em diferentes qualidades e segmentação dos vídeos em pequenos fragmentos de curta duração, faz-se necessário a geração do arquivo de manifesto, que conterá informações sobre as representações do vídeo nas qualidades disponíveis (*bitrates*) bem como informações de localização dos segmentos e o tempo de duração dos mesmos, dentre outras informações técnicas como tempo total de duração do vídeo e a resolução disponível. O arquivo de manifesto possui as informações de localização dos segmentos de vídeo em diferentes

representações, acessíveis por meio do protocolo *HTTP*. A transferência do vídeo adaptativo ocorre através de requisições e respostas, utilizando o protocolo os métodos *HTTP Request* e *HTTP Response* [26] do protocolo *HTTP*.

Baseado nas informações descritas no arquivo de manifesto, o cliente faz a solicitação do segmento inicial e dos próximos segmento de vídeo ao servidor, que por sua vez fornece o segmento com a qualidade mais apropriada para o determinado instante de transmissão e assim sucessivamente. Durante a transmissão do vídeo, conforme variação da largura de banda, a cada requisição, o cliente seleciona a representação mais adequada. Conforme mecanismo de estimação, que se baseia na largura de banda disponível, os próximos segmentos serão obtidos nas qualidades que mais se aplicam às condições da rede no determinado instante.

Finalizando a etapa de preparação de conteúdo de vídeo, os arquivos referentes aos segmentos, juntamente com um arquivo de segmento inicial e o arquivo de manifesto, são organizados em suas respectivas estrutura de pastas no servidor de vídeo. Sendo assim, o servidor de vídeo disponibiliza a URL (*Uniform Resource Locator*) de acesso ao arquivo de manifesto para o cliente que por sua vez deve possuir um *player* compatível com o padrão MPEG-DASH para realizar a requisição e interpretação do arquivo de manifesto e posteriormente iniciar a reprodução do conteúdo de vídeo adaptativo.

5.1.2 Ambiente Computacional

A etapa de configurar o ambiente computacional envolve a preparação do *hardware* e dos *softwares* bem como o desenvolvimento dos diversos *scripts* necessários à implementação da estrutura. A fim de transmitir fluxos reais de vídeo pelo simulador NS-3, foram implantadas duas máquinas virtuais com sistema operacional Linux, sendo uma para a instalação do servidor de vídeo e outra para a instalação do cliente.

Para suportar o recurso de máquinas virtuais, foi utilizado o gerenciador de máquinas virtuais *VirtualBox*. Na plataforma do hospedeiro foi instalado o simulador NS-3 [45] provendo assim os recursos necessários para implementação de uma rede IP simulada para transmissão do vídeo.

O NS-3 é um simulador de código aberto, baseado em eventos discretos, disponível primordialmente para pesquisa além de permitir desenvolvimento de modelos realísticos e conexões com redes reais [45]. O simulador de rede NS-3 permite o desenvolvimento de simulações de diversas topologias de rede, inclusive da pilha de protocolos TCP/IP, possuindo suporte às tecnologias AQM, sendo possível modificar o algoritmo de gerenciamento de fila dos ativos de rede.

Para que a conexão entre servidor e cliente, passando pela rede simulada, seja possível, é necessário, antes de tudo, prover a comunicação das máquinas virtuais com o sistema operacional hospedeiro. Sendo assim, visando a comunicação entre os nós reais e os nós simulados, ou seja, a comunicação entre as máquinas virtuais e os dispositivos da rede simulada pelo NS-3, foi implementado o recurso conhecido por *TAP Net Device* com a configuração no modelo *TapBridge Model*, utilizando o modo de operação *UseBridge Mode* [45].

A implementação de recursos de *TAP* e *Bridge* no NS-3 é realizada por meio do módulo *Tap Bridge Network Device* [45] que fornece uma API (*Application Program Interface*) para interconexão através de recursos de IPC (*Inter Process Communication*) do sistema operacional em que o simulador encontra-se instalado. Dessa forma foram criadas interfaces de rede virtuais do tipo *TAP* no sistema operacional hospedeiro que irão simular novas interfaces de rede em conjunto com barramentos lógicos do tipo *bridge* para replicar os pacotes entre essas interfaces.

Através das *bridges*, os pacotes que chegam são redirecionados para as interfaces do sistema operacional convidado e vice-versa, replicando o pacote para os endereços IP associados às máquinas virtuais. Com a interconexão implementada, as duas máquinas virtuais foram configuradas e conectadas aos respectivos roteadores simulados no NS-3. A Figura 5.2 [45] ilustra o esquema de operação do módulo *Tap Bridge Network Device* no NS-3.

No sistema operacional das duas máquinas virtuais, foram configurados todos os softwares e ferramentas necessárias ao servidor de vídeo e respectivo cliente, de forma a possibilitar a transmissão do vídeo adaptativo no padrão DASH, além da gravação do conteúdo recebido pelo cliente. A transmissão e recebimento do vídeo foi realizada por meio de aplicações disponibilizadas pelo Framework Multimedia GPAC (*Multimedia Open Source Project*) [54].

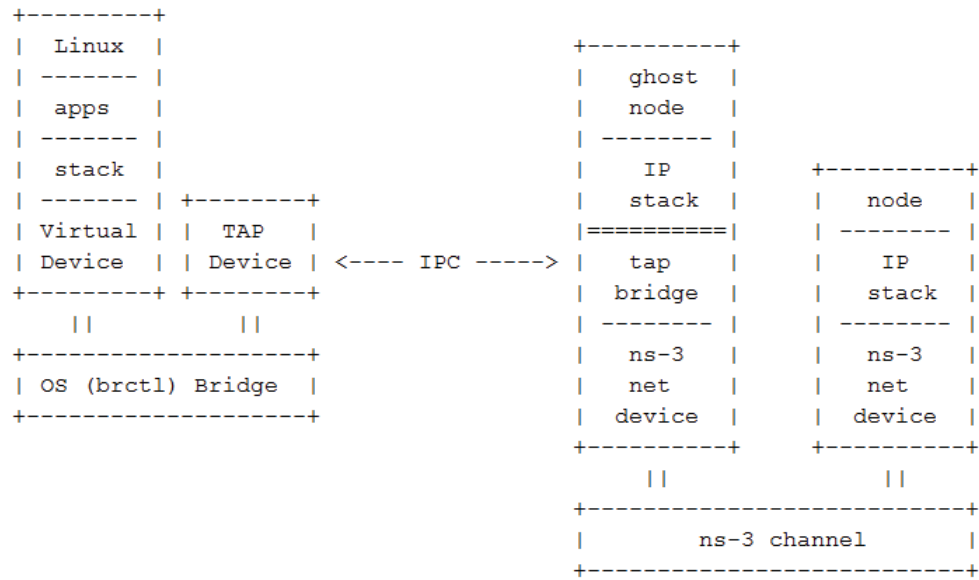


Figura 5.2: Esquema de operação do módulo *Tap Bridge Network Device* no simulador NS-3.

5.2 Implementação das Simulações

Durante a etapa de implementação dos *scripts* de simulação no NS-3, algumas tarefas são essenciais para que o método proposto possa ser validado com consistência, nas quais encontram-se a definição e parametrização dos cenários, configuração da fila de pacotes a ser monitorada além do esquema de geração de tráfego concorrente.

5.2.1 Definições de Rede

Na implementação da rede simulada, foi utilizada a topologia *dumbbell*, tradicionalmente utilizada na avaliação de desempenho de algoritmos para gerenciamento ativo de filas, como pode-se observar no estudo de *Subramanya et al.* [15]. Foi realizada a implementação dos *scripts* de simulação da rede IP contendo dois roteadores principais para formação de um enlace de gargalo. Os roteadores são interconectados por um *link* com largura de banda de 10 Mbps e latência de 5 ms. Os demais enlaces de acesso possuem largura de banda de 10 Gbps e latência de 1 ms.

O enlace principal interliga os roteadores que promovem o acesso às fontes transmissoras e aos clientes receptores do tráfego mediante os enlaces de acesso. O roteador principal recebe

o tráfego oriundo do fluxo de vídeo adaptativo bem como das outras fontes responsáveis pela geração dos fluxos concorrentes. O roteador secundário recebe o tráfego do link principal e faz o roteamento e entrega do fluxo de vídeo adaptativo ao cliente, realizando também a entrega dos fluxos concorrentes para os destinos correspondentes.

Quanto à capacidade do enlace de gargalo, que foi fixada em 10 Mbps, buscou-se simular os enlaces destinados para acessos domésticos de Internet, que apesar de muitas vezes possuírem largura de banda maior especificada, na prática são compartilhados por diversos dispositivos como smartphones e aparelhos eletrônicos, resultando em menor capacidade disponível para receber o *streaming* de vídeo. A largura de banda de 10 Gbps referente aos demais enlaces de acesso, tem por objetivo gerar requisições e respostas em velocidade superior ao enlace de gargalo, reproduzindo os efeitos do congestionamento no *link* principal. A Figura 5.3 apresenta o cenário implementado para as simulações.

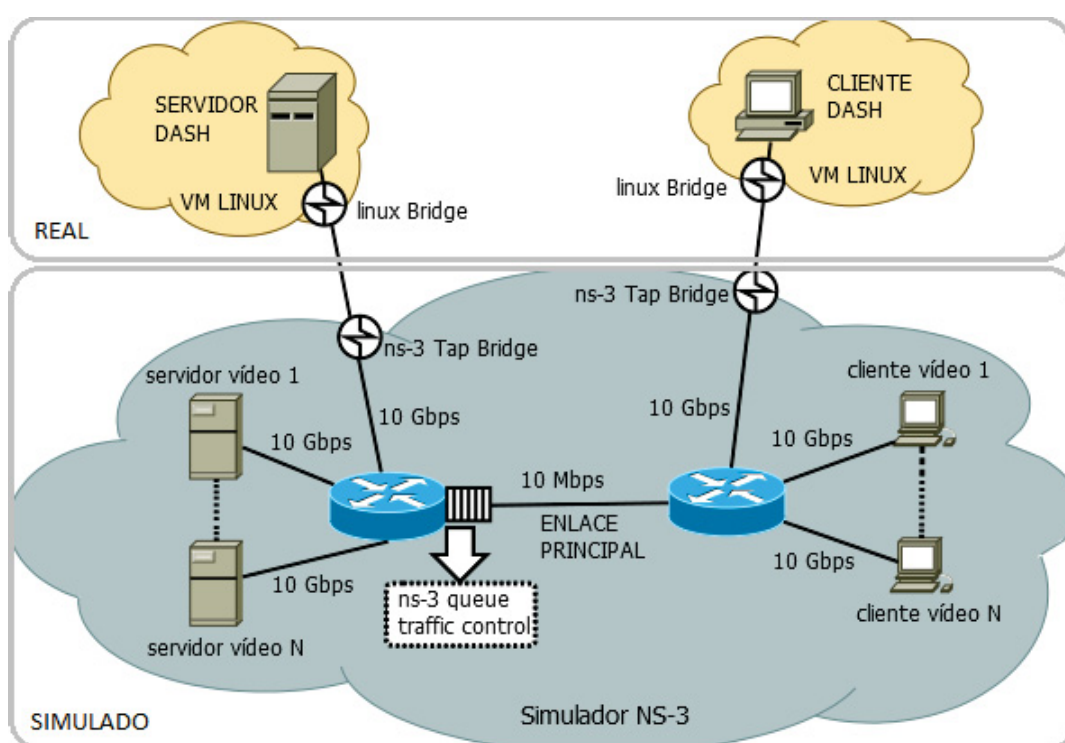


Figura 5.3: Configuração para análise de desempenho dos métodos AQM.

5.2.2 Tráfego Concorrente

Foram implementadas no simulador fontes geradoras de tráfego TCP para competir por recursos da rede com o servidor de vídeo de fluxo adaptativo, provocando assim o congestionamento no enlace principal na medida em que se aumenta a utilização do canal de comunicação. A fim de inserir tráfego de fundo no cenário, buscando prover gradativamente o congestionamento no link de gargalo por meio da elevação do nível de utilização do enlace pelas fontes concorrentes, 10 fontes geradoras de tráfego e seus respectivos clientes transmitem pacotes de 1500 *bytes* sobre o protocolo TCP, com intervalos de geração variáveis ao longo das simulações.

A geração de tráfego concorrente segue o modelo PPBP (*Poisson Pareto Burst Process*) [56]. O modelo PPBP representa de forma realística a agregação de tráfego na Internet, no qual sua distribuição baseia-se em modelos LRD (*Long-Range Dependence*). Os resultados obtidos

no estudo de *Ammar et al.* [56] comprovam as características LRD através de plotagem da função de auto-correlação.

Baseado na sobreposição de múltiplas rajadas de pacotes, o modelo PPBP possui características de auto-similaridade e cauda pesada. As rajadas chegam de acordo com um processo de *Poisson* e seu comprimento segue uma distribuição de *Pareto* [56]. O parâmetro de *Hurst* foi configurado em 0.7, o que é compatível com o tráfego gerado por aplicações de vídeo [57]. Os demais parâmetros são setados conforme a simulações, aumentando a intensidade de tráfego gerado pelas fontes concorrentes de acordo com a observação do nível de utilização do enlace principal no cenário implementado.

5.2.3 Dinâmica dos Experimentos

Após implementação do cenário de rede IP e respectivos nós e roteadores, o método AQM a ser avaliado é implementado na fila do roteador principal. O roteador principal recebe os pacotes das fontes de tráfego e as transfere para o enlace de gargalo, que por sua vez possui menor largura de banda e maior atraso que os enlaces de acesso. Gradativamente intensifica-se o nível de geração de tráfego concorrente. Em momentos de elevados níveis de utilização no enlace principal, com recebimento de tráfego em rajadas, há um grande número de pacotes chegando ao *buffer* do roteador para serem enfileirados. Consequentemente, o método AQM efetua descartes conforme técnica implementada e características de comportamento do algoritmo. Conforme o nível de congestionamento aumenta, impulsionado pelo maior volume de tráfego gerado pelas fontes concorrentes, o número de pacotes descartados também tende a aumentar, alterando o comportamento das fontes de tráfego que utilizam o protocolo de transporte TCP, que por sua vez diminuem suas taxas de transmissão. Com isso é possível realizar a análise de impacto na alteração de qualidade do fluxo de vídeo adaptativo, que busca transferir os segmentos de vídeo que se enquadram na melhor qualidade (*bitrate*) correspondente à largura de banda disponível.

Foram conduzidos extensos experimentos iniciais para compreender os efeitos dos parâmetros na configuração do cenário. Diversas simulações foram realizadas, variando-se o método AQM na fila do roteador no gargalo da rede e intensificando o nível de utilização do enlace principal por meio do aumento das taxas de transmissão das fontes de tráfego de fundo. Para cada vídeo e intensidade de tráfego, além do método proposto, foram testados os seguintes métodos AQM conhecidos: RED, ARED, CoDel e PIE. Os métodos AQM foram configurados contendo um *buffer* com capacidade máxima de 500 pacotes na fila. O tamanho da fila foi fixado em 500 pacotes, valor relativamente baixo para os roteadores atuais, com o objetivo de que essa capacidade seja rapidamente esgotada. Dessa forma é possível representar os efeitos do descartes no *buffer* em um curto período de tempo. Uma vez que filas com capacidade maiores necessitam de mais tráfego de fundo e mais tempo de vídeo para que se fosse possível perceber as ocorrências de descartes de pacotes.

Também foi realizada a avaliação em comparação com o algoritmo DropTail, utilizando um *buffer* com capacidade de 2 milhões de pacotes, suficiente para não ocorrer descartes de pacotes. Como o DropTail realiza descarte somente quando a fila é preenchida em sua capacidade máxima, é possível dessa forma simular um *buffer* infinito, provocando os efeitos do *Bufferbloat*. Para o cenário implementado, a capacidade de 2 milhões de pacotes, levando-se em conta o tamanho dos vídeos e o número de fontes concorrentes, trata-se de um valor representativo para se perceber, mesmo em menor escala, o efeito do *Bufferbloat* na transmissão de vídeo adaptativo.

Sendo assim, durante a realização dos experimentos, os métodos AQM e o DropTail foram comparados em simulações distintas envolvendo os mesmos cenários e o mesmos percentuais de utilização do enlace principal. A cada rodada de experimentos foi intensificada a geração de

tráfego concorrente, o que reflete diretamente no nível de ocupação do enlace principal, variando de intensidade entre uma taxa de ocupação moderada, de 60%, aumentando gradativamente até a máxima taxa ocupação simulada, de 99%. A taxa de ocupação do enlace principal levou em consideração somente o tráfego gerado pelos fluxos concorrentes, sendo que a largura de banda restante da ocupação era destinada ao fluxo de vídeo adaptativo. As simulações consideram que os vídeos requisitados são assistidos pelo usuário de forma contínua em sua totalidade, sem interrupções ou pausas intencionais.

5.2.4 Parametrização dos Métodos AQM

O método proposto foi implementado no simulador NS-3 por meio de alterações no modelo implementado do RED na classe *Traffic Control Layer RED queue disc* [45], realizando uma conversão direta entre o número de pacotes na fila e tempo estimado para permanência dos mesmos, por meio da Lei de *Little* [46]. Os demais métodos AQM avaliados foram utilizados em sua configuração padrão conforme implementações na *Model Library: Traffic Control layer - Queue disciplines* [45] existente na versão 3.29 do simulador de rede NS-3. A tabela 5.4 apresenta a configuração utilizada nos métodos avaliados.

5.3 Análises de Resultados

Para análise dos resultados, buscando mensurar de forma objetiva a qualidade do vídeo transmitido, foi realizado o cálculo do PSNR médio do vídeo recebido em comparação com o original, além do atraso médio e o *jitter* dos fluxos concorrentes.

5.3.1 Peak Signal to Noise Ratio - PSNR

O PSNR (*Peak Signal to Noise Ratio*) consiste em uma métrica objetiva para avaliação da qualidade de vídeo, que avalia a similaridade entre duas imagens, computando o erro médio quadrático MSE (Mean Square Error) de cada pixel entre o original e as imagens recebidas. O PSNR médio é obtido por meio da avaliação do resultado quadro a quadro, de forma que quando finalizados todos os quadros é calculada uma média aritmética para compor o resultado final [58]. A unidade utilizada como medida para PSNR é o dB (decibel). Quanto maior o PSNR, melhor a qualidade percebida. Sendo assim, imagens com mais similaridade resultam em valores mais altos de PSNR [59]. Ou seja, quanto mais próxima a imagem transmitida em relação da recebida, maior será o valor obtido pelo PSNR médio. O cálculo do MSE é dado por:

$$MSE = \frac{1}{rc} \sum_{i=1}^r \sum_{j=1}^c [X_o(i, j) - X_r(i, j)]^2 \quad (5.1)$$

onde r e c representam respectivamente o número de linhas e colunas da imagem; e, $X_o(i, j)$ e $X_r(i, j)$ representam os valores obtidos para o *pixel* (i, j) do quadro original e do quadro recebido, respectivamente. Após calculado o MSE, o PSNR é obtido por:

$$PSNR = 20 \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \quad (5.2)$$

onde MAX_I representa o valor máximo da intensidade do *pixel*. Para os vídeos utilizados nesse trabalho, considera-se $MAX_I = 255$.

Tabela 5.4: Configuração dos Métodos Avaliados.

AQM	Configuração e Parâmetros
DropTail	Fila com capacidade máxima de 2 milhões de pacotes. Módulo <i>Traffic Control Layer FifoQueueDisc</i> , <i>MaxSize</i> =2000000. Os demais parâmetros permanecem em sua configuração padrão disponível na versão 3.29 do NS-3.
RED	Fila com capacidade máxima de 500 pacotes. Limites de tamanho médio da fila: mínimo 250 pacotes e máximo 500 pacotes. Módulo <i>Traffic Control Layer RED queue disc</i> , <i>MaxSize</i> =500, <i>MinTh</i> =250, <i>MaxTh</i> =500. Os demais parâmetros permanecem em sua configuração padrão disponível na versão 3.29 do NS-3: <i>Queue weight</i> =0.002, <i>max_p</i> = 1/50
ARED	Fila com capacidade máxima de 500 pacotes. Limites de tamanho médio da fila: mínimo 250 pacotes e máximo 500 pacotes. Módulo <i>Traffic Control Layer RED queue disc</i> , <i>ARED</i> = true, <i>MaxSize</i> =500, <i>MinTh</i> =250, <i>MaxTh</i> =500. Os demais parâmetros permanecem em sua configuração padrão disponível na versão 3.29 do NS-3: <i>Queue weight</i> =0.002, <i>max_p</i> = 1/50, $\alpha = 3$, $\beta = 2$.
CoDel	Fila com capacidade máxima de 500 pacotes. Módulo <i>Traffic Control Layer CoDel queue disc</i> , <i>MaxSize</i> =500. Os demais parâmetros permanecem em sua configuração padrão disponível na versão 3.29 do NS-3: <i>Interval</i> : 100 ms, <i>Target</i> : 5 ms.
PIE	Fila com capacidade máxima de 500 pacotes. Módulo <i>Traffic Control Layer PIE queue disc</i> , <i>MaxSize</i> =500. Os demais parâmetros permanecem em sua configuração padrão disponível na versão 3.29 do NS-3: <i>Tupdate</i> : 30 ms, <i>DequeueThreshold</i> : 10000 bytes, <i>QueueDelayReference</i> : 20 ms, <i>MaxBurstAllowance</i> : 0.1 seconds, <i>alpha</i> : 0.125, <i>beta</i> : 1.25..
Método proposto	Fila com capacidade para 500 pacotes. Limites de tempo de permanência mínimo de 100ms e máximo de 200ms. Módulo <i>Traffic Control Layer RED queue disc</i> , <i>MaxSize</i> =500, com alterações e conversão de número de pacotes para tempo baseado na <i>Lei de Little</i> .

MOS (*Mean Opinion Score*) [43] é uma medida subjetiva de qualidade de experiência do usuário, que fornece uma indicação numérica da qualidade da mídia recebida, onde 1 corresponde ao pior valor e 5 ao melhor valor de qualidade possível. No entanto, métodos subjetivos como MOS têm alto custo para serem realizados, uma vez que há o envolvimento de diversas pessoas e equipamentos, fazendo com que os requisitos para implementar o ambiente de teste sejam bastante complexos [53]. Por outro lado, métodos objetivos como o PSNR usam ferramentas computacionais e abordagem estatística avaliar a qualidade da mídia recebida. A tabela 5.5 [53] apresenta a relação entre os valores de PSNR e a medida de qualidade MOS que classifica subjetivamente em Excelente, Bom, Razoável, Pobre e Ruim, o resultado da qualidade recebida de acordo com o valor do PSNR obtido. Para este trabalho, o uso do PSNR é justificado porque as comparações são realizadas com o base na mensuração objetiva de maior ou menor redução da qualidade entre o vídeo original transmitido e o vídeo recebido, buscando retratar a qualidade percebida pelo usuário. Sendo assim, a qualidade absoluta medida com o MOS não se faz necessária, pois o presente trabalho não possui foco em avaliação subjetiva referente à qualidade de experiência do usuário.

Tabela 5.5: Relação entre PSNR e MOS.

PSNR (dB)	MOS	Significado
$\text{PSNR} > 37$	5	Excelente
$31 < \text{PSNR} \leq 37$	4	Bom
$25 < \text{PSNR} \leq 31$	3	Razoável
$20 < \text{PSNR} \leq 25$	2	Pobre
$\text{PSNR} \leq 20$	1	Ruim

5.3.2 Padrão Ouro

Com o objetivo de conhecer o valor máximo de PSNR médio para cada vídeo, foram realizadas simulações para compor o Padrão Ouro. Nessa configuração há a transmissão dos vídeos sem fontes de tráfego concorrente. Em tais simulações, como não há congestionamento, consequentemente não há descartes de pacotes. Com isso, o vídeo é entregue na melhor qualidade, ou seja, na maior representação. O valor do PSNR médio máximo também leva em consideração as perdas que ocorrem no momento da codificação para o padrão H.264/AVC. A tabela 5.6 apresenta o resultado do PSNR médio máximo obtido para o Padrão Ouro de cada vídeo.

Tabela 5.6: PSNR médio Máximo - Padrão Ouro.

Vídeo	PSNR médio (dB)
<i>Big Buck Bunny</i>	37.0
<i>Sunflower</i>	38.8
<i>Rush Hour</i>	40.4
<i>Touchdown Pass</i>	40.3
<i>Pedestrian Area</i>	39.6
<i>Riverbed</i>	30.7

5.3.3 Atraso Médio e Jitter

Buscando analisar outras métricas de desempenho para os métodos AQM avaliados, também foram coletadas estatísticas de tráfego para observação e análise de resultados envolvendo o atraso médio e o *jitter*. Os valores estatísticos foram coletados e mensurados durante as simulações através da classe *Flow Monitor* do simulador NS-3 [45]. O atraso é o tempo que um pacote leva para atravessar uma rede, passando pelos roteadores e enlaces intermediários, desde a origem até o destino. O atraso total fim a fim será a soma dos atrasos em cada um dos nós do caminho. O atraso médio é medido em termos de tempo, geralmente em milissegundos (*ms*) ou segundos (*s*). Atrasos de poucos milissegundos costumam ser desprezíveis, por outro lado um atraso de alguns segundos pode ser bastante significativo dependendo da distância entre a origem e o destino. Os pacotes de uma determinada fonte chegam ao destino com diferentes tempos de atraso. O *jitter* indica a variação do atraso de ida e volta de um pacote entre a origem e o destino, podendo ser definido como sendo a variação estatística do atraso entre os pacotes sucessivos na entrega de dados em uma rede [3]. Uma variação de atraso elevada produz uma recepção não regular dos pacotes. A alta variação no atraso pode afetar a qualidade de um *streaming* de vídeo [40].

Capítulo 6

Resultados

6.1 PSNR Médio

Após coleta e tratamento dos dados resultantes das simulações foram gerados gráficos comparativos entre os métodos AQM testados. A qualidade do vídeo recebido foi estimada com o cálculo do PSNR médio entre o vídeo original e o recebido pelo cliente. Observa-se que conforme o nível de utilização aumenta, há diferença entre o PSNR obtido, o que impacta diretamente na qualidade do vídeo recebido. Quanto maior o PSNR, melhor a qualidade percebida. A análise de resultados foi realizada por meio da comparação dos resultados entre os métodos AQM em função do nível de ocupação do enlace principal.

Em um primeiro experimento foram utilizados os primeiros 60 segundos do vídeo *Big Buck Bunny*. Este vídeo possui o maior tempo de duração dentre os selecionados para os testes, porém possui a menor taxa de quadros por segundo, com 24 *Frames* por segundo. Por se tratar de um desenho, o vídeo proporciona uma maior compactação espacial e temporal. A Figura 6.1, mostra que a medida em que se intensifica o nível de ocupação do enlace principal, se amplia a distância entre o PSNR médio resultante entre os métodos testados. Pode-se perceber expressiva vantagem do método proposto a partir do nível 70% de utilização do enlace principal. Constata-se também que o DropTail apresentou pior desempenho com uma queda acentuada na qualidade após 70% de utilização. Os demais métodos proporcionam qualidades intermediárias. O padrão ouro para o vídeo *Big Buck Bunny* é de 37.0064 dB. Observa-se que somente em 60% de utilização os métodos alcançam resultados próximos ao padrão ouro, pois após 70% os resultados começam a se distanciar do máximo valor possível.

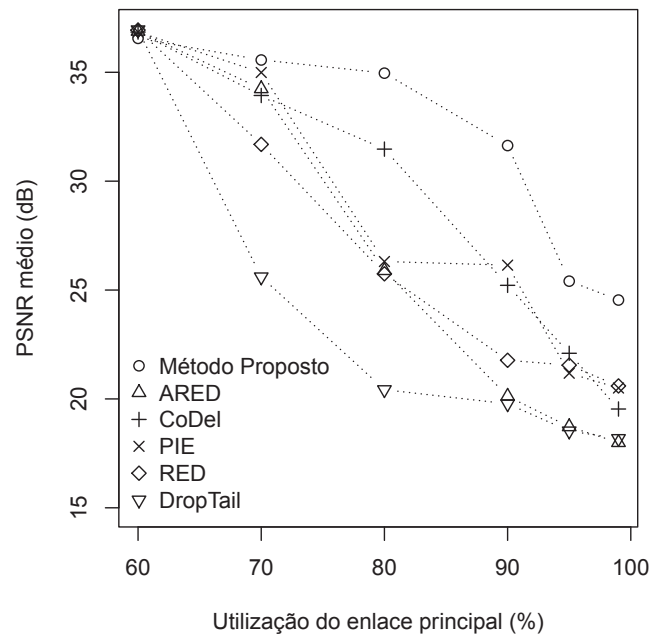


Figura 6.1: PSNR médio resultante para o vídeo *Big Buck Bunny*.

Em um segundo experimento foi realizada a simulação da transmissão do vídeo *Rush Hour* que é constituído de cenas reais de intenso tráfego de veículos e ondas de calor, o que tende diminuir a compressão. Este vídeo possui um tempo de duração de 20 segundos, sendo intermediário dentre os selecionados para os testes. Possui taxa de 25 *Frames* por segundo. De acordo com o resultado apresentado na Figura 6.2 nota-se que, com exceção do DropTail que produz maior queda na qualidade, todos os demais métodos AQM produzem resultados próximos até o nível de 80% de utilização do enlace principal. O PIE e o RED apresentaram degradação grande de qualidade em 95% de utilização. O CoDel, ARED e o método proposto apresentaram desempenho semelhante. O padrão ouro para o vídeo *Rush Hour* é de 40.4929 dB. Em 60% e 70% de utilização do enlace principal todos os métodos apresentam resultados bastante próximos ao padrão ouro.

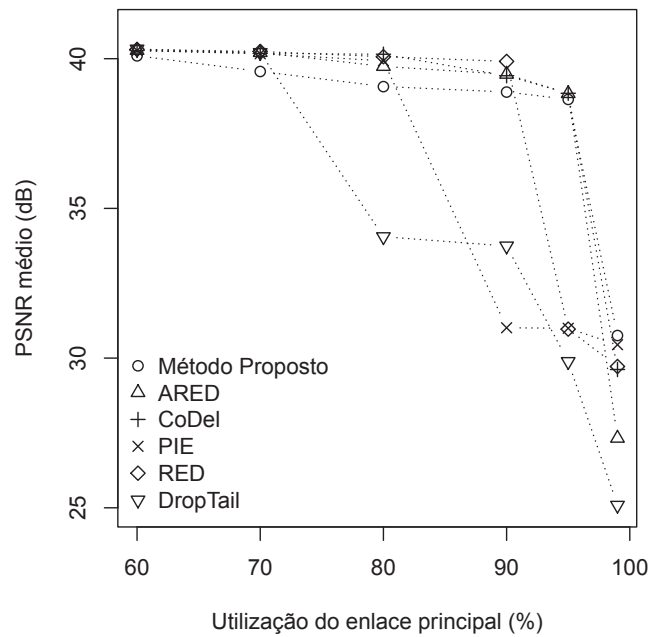


Figura 6.2: PSNR médio resultante para o vídeo *Rush Hour*.

No terceiro experimento foi utilizado o vídeo *Sunflower*, que é composto por cenas com pouca variação de movimento com uma abelha no centro e um girassol ao fundo. Contendo 20 segundos de duração, este vídeo possui um tempo intermediário dentre os selecionados para os testes. Sua taxa de amostragem é de 25 *Frames* por segundo. Através da Figura 6.3 percebe-se que o PSNR médio foi semelhante até o nível de 80% de utilização do enlace principal. Novamente o Droptail apresentou pior desempenho. O CoDel apresentou degradação na qualidade a partir de 80% de utilização do enlace principal e o PIE e ARED a partir de 90%. Após 90% observa-se que o método proposto produz maior qualidade. O padrão ouro para o vídeo *Sunflower* é de 38.8591 dB. Todos os métodos obtêm resultados próximos ao padrão ouro em 60% e 70% de utilização do enlace principal.

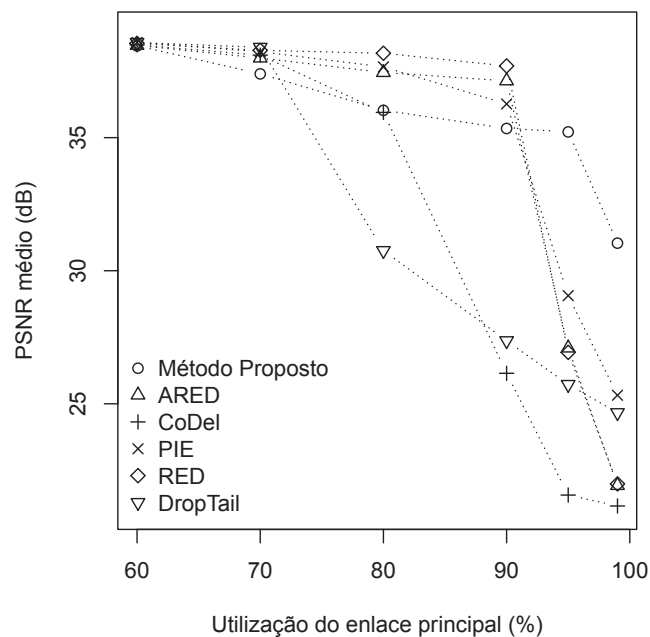


Figura 6.3: PSNR médio resultante para o vídeo *Sunflower*.

Em um quarto experimento, utilizou-se o vídeo *Touchdown Pass* que possui cenas de um jogo de futebol americano, promovendo um padrão de movimento bastante intenso com rápidas transições entre as mesmas. Contendo 19 segundos de duração, este vídeo possui um tempo intermediário dentre os selecionados para os testes, porém apresenta maior taxa de quadros por segundo em relação aos demais, com 30 *Frames* por segundo. De acordo com a Figura 6.4, a partir do nível de utilização de 90% do enlace principal, também observa-se o benefício da utilização do método proposto. O PIE apresentou degradação na qualidade a partir de 80% de utilização enquanto que os demais métodos degradaram mais intensamente a qualidade a partir de 95% de utilização. Novamente o DropTail apresentou o pior desempenho. O padrão ouro para o vídeo *Touchdown Pass* é de 40.3951 dB. Observa-se que todos os métodos proporcionam resultados próximos ao padrão ouro em 60% e 70% de utilização do enlace principal.

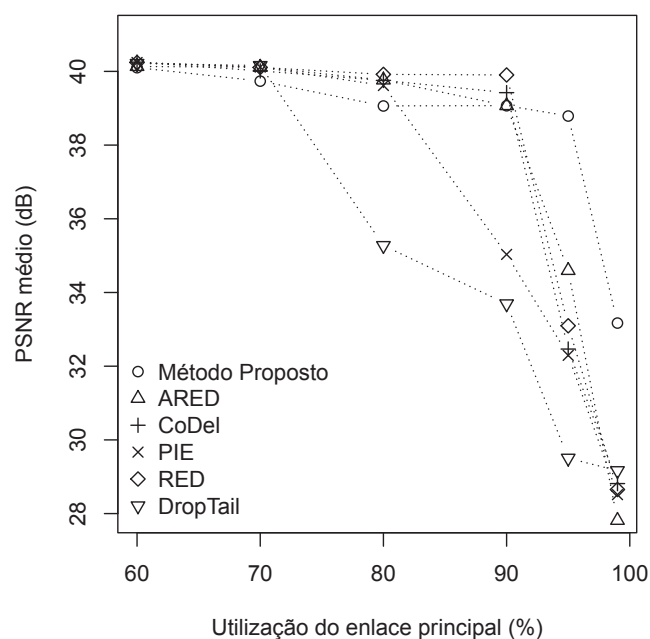


Figura 6.4: PSNR médio resultante para o vídeo *Touchdown Pass*.

No quinto experimento, foi utilizado o vídeo *Pedestrian Area* que consiste em cenas reais de intenso tráfego de pedestres em um centro urbano. Contendo 15 segundos de duração, este vídeo possui o segundo menor tempo dentre os selecionados para os testes. Possui taxa de 25 *Frames* por segundo. Conforme pode-se observar na Figura 6.5, a partir de 90% de utilização do enlace principal, quando se intensifica o congestionamento, o Método Proposto ganha vantagem, gerando maiores resultados de PSNR médio. Na sequência do Método Proposto, encontram-se os métodos CoDel e pelo PIE, com resultados bastante próximos em 95% de utilização do enlace principal. O Droptail continua levando o pior desempenho, pois a partir de 80% de utilização do enlace principal apresenta uma grande queda de desempenho. O RED e o ARED apresentam resultados intermediários, tendo performance quase que idêntica em 80% de utilização do enlace principal. O padrão ouro para o vídeo *Pedestrian Area* é de 39.6138 dB. Constata-se que em 60% e 70% de utilização do enlace principal todos os métodos apresentam resultados próximos ao padrão ouro.

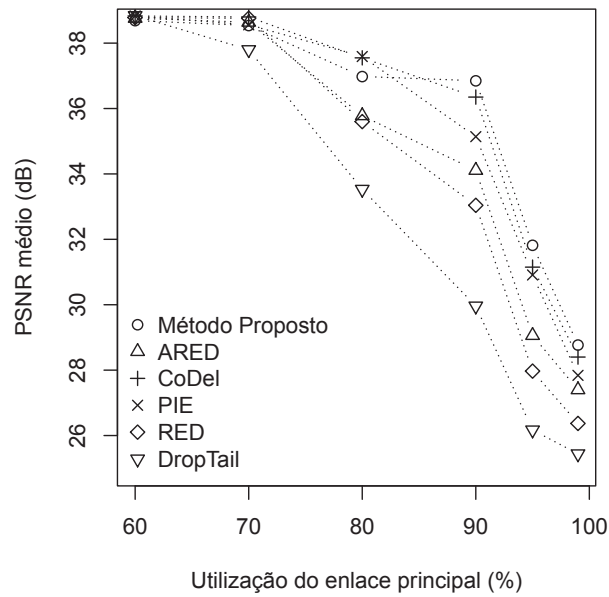


Figura 6.5: PSNR médio resultante para o vídeo *Pedestrian Area*.

Em um sexto experimento foi realizada a transmissão do vídeo *Riverbed* que é composto basicamente de cenas de uma nascente de água, contendo bastante movimento provocado pelas ondas na superfície em meio aos reflexos da luz. Com apenas 10 segundos de duração, este vídeo possui o menor tempo dentre os selecionados para os testes. Sua taxa é de 25 *Frames* por segundo. Através da Figura 6.6 pode-se constatar que o Método Proposto mostra sua eficiência, em relação aos demais métodos, a partir do nível de utilização de 90% do enlace principal. Novamente o CoDel e PIE apresentam os melhores resultados depois do Método Proposto. Droptail continua com expressiva desvantagem a partir de 80% de utilização do enlace principal. RED e ARED apresentam resultados bastante próximos, distanciando apenas em 95% de utilização. O padrão ouro para o vídeo *Riverbed* é de 30.7509 dB. Percebe-se que os métodos alcançam resultados próximos ao padrão ouro apenas em 60% e 70% de utilização.

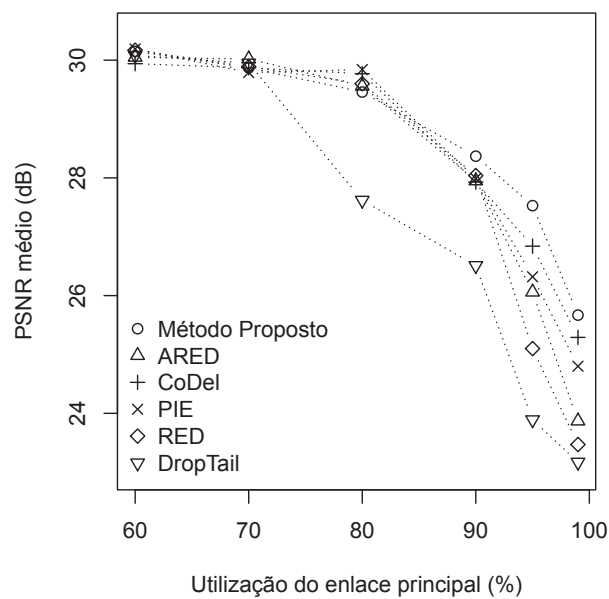


Figura 6.6: PSNR médio resultante para o vídeo *Riverbed*.

6.2 Atraso Médio e *Jitter*

Uma outra etapa de análise de resultados diz respeito à observação do atraso médio e *jitter* dos fluxos concorrentes. Os fluxos concorrentes são compostos por todos os pacotes gerados pelas demais fontes TCP que disputam os recursos disponíveis com o tráfego de vídeo adaptativo. A análise foi realizada por meio da coleta de estatísticas de tráfego durante as simulações de transmissão dos vídeos em função do nível de utilização do enlace principal.

O atraso é o tempo absoluto que um pacote leva para ser transmitido de uma rede de origem a uma rede de destino, enquanto que o *jitter* é uma medida da variação do atraso [60]. Atrasos de entregas de pacotes em uma rede podem ser ocasionados por diversos fatores, sendo que o congestionamento favorece o aumento do atraso devido ao tempo que o pacote permanece na fila aguardando para ser transmitido. O *jitter*, por sua vez, constitui a variação estatística do atraso na entrega dos pacotes aos destinos.

Inicialmente, a Figura 6.7 apresenta o resultado do atraso médio para os fluxos concorrentes durante a transmissão dos primeiros 60 segundos do vídeo *Big Buck Bunny*. Pela observação do gráfico pode-se perceber expressiva vantagem do método proposto com menor atraso médio para todos os níveis de utilização do enlace principal. Consta-se também que o DropTail apresentou pior desempenho com maior atraso médio para todos os níveis de utilização, seguido pelo RED que apresenta segundo pior desempenho. Os demais métodos AQM PIE, CoDel e ARED proporcionam resultados intermediários.

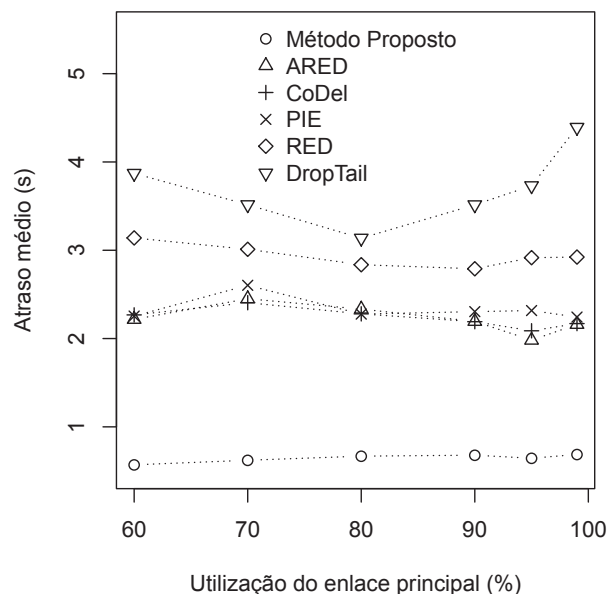


Figura 6.7: Atraso médio resultante para o vídeo *Big Buck Bunny*.

Conforme a Figura 6.8, na transmissão do vídeo *Big Buck Bunny* o método AQM proposto possibilita menor *jitter* para todos os níveis de utilização do enlace principal. Os demais métodos AQM avaliados apresentaram resultados bastante próximos, com valores mais altos para o *jitter* em todos os níveis de utilização do enlace principal.

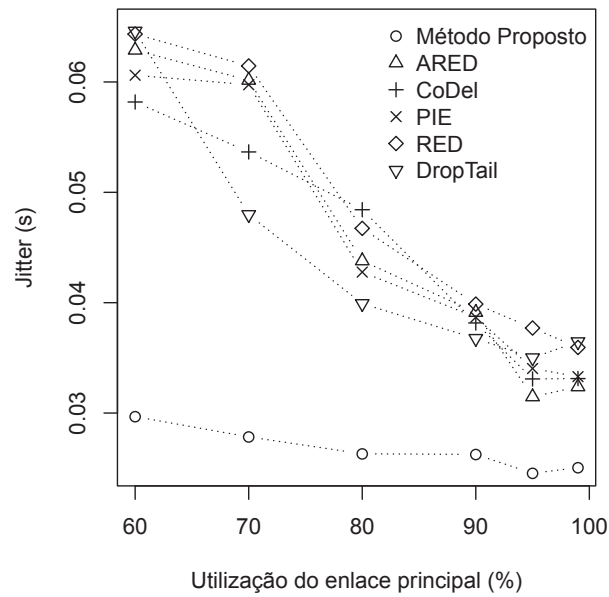


Figura 6.8: Jitter resultante para o vídeo *Big Buck Bunny*.

Na sequência, é exposto o resultado do atraso médio para os fluxos concorrentes durante a transmissão do vídeo *Rush Hour*. Conforme a Figura 6.9, nota-se que novamente o método proposto proporciona vantagem resultando em menor atraso médio para todos os níveis de utilização do enlace principal. O DropTail volta a apresenta o pior desempenho com maior atraso médio para todos os níveis de utilização, seguido pelo RED. Resultados intermediários são obtidos pelos demais métodos AQM avaliados.

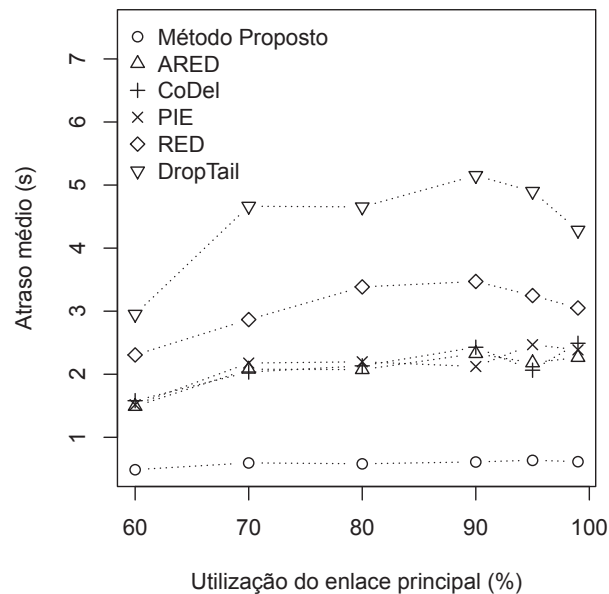


Figura 6.9: Atraso médio resultante para o vídeo *Rush Hour*.

Em seguida, percebe-se novamente a vantagem do método AQM proposto na transmissão do vídeo *Rush Hour*. De acordo com a Figura 6.10 observa-se que o método proposto proporciona menor *jitter* para todos os níveis de utilização do enlace principal. O RED apresenta maior *jitter* em quase todos os níveis de utilização, produzindo pior desempenho. Os métodos PIE, CoDel e ARED resultam em valores intermediários de *jitter* para os níveis de utilização simulados.

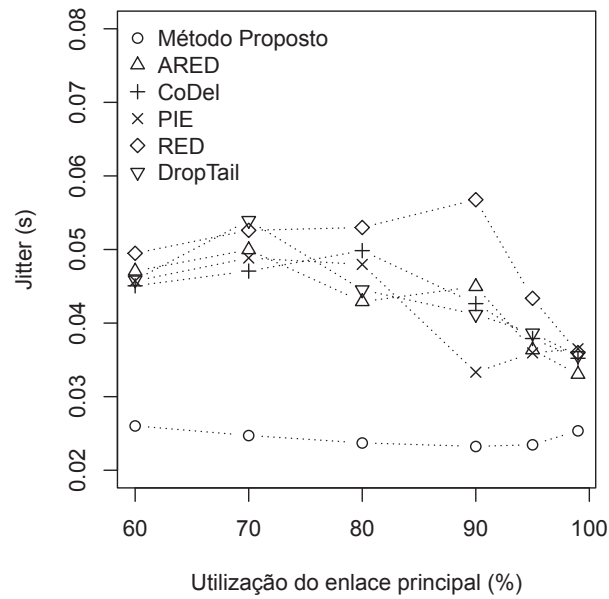


Figura 6.10: Jitter resultante para o vídeo *Rush Hour*.

A Figura 6.11 mostra o resultado do atraso médio para os fluxos concorrentes durante a transmissão do vídeo *Sunflower*. A vantagem do método AQM proposto é constatada novamente devido aos menores valores para obtidos para o atraso médio em todos os níveis de utilização do enlace principal. O pior desempenho é novamente do DropTail que proporciona maiores valores para o atraso médio para todos os níveis de utilização, novamente seguido pelo RED. Os outros métodos AQM PIE, CoDel e ARED permanecem com resultados intermediários.

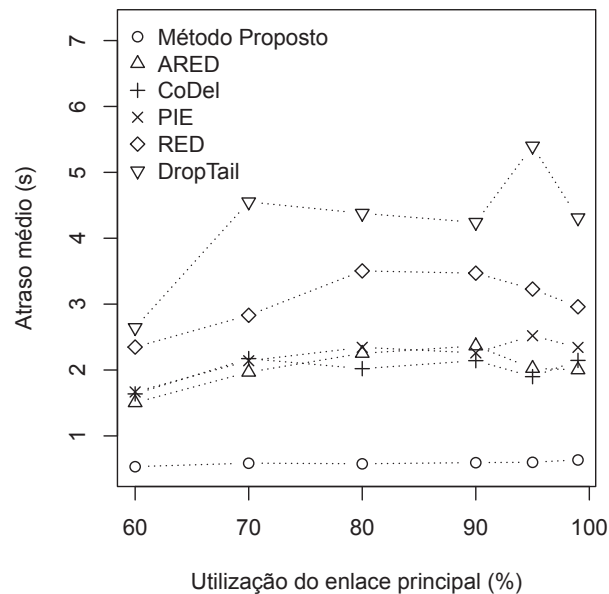


Figura 6.11: Atraso médio resultante para o vídeo *Sunflower*.

Na sequência a Figura 6.12 apresenta os resultados obtidos na transmissão do vídeo *Sunflower*. Constata-se novamente que o método AQM proposto produz um menor *jitter* para todos os níveis de utilização do enlace principal. O RED mantém o pior desempenho em quase todos os níveis de utilização, resultando em maior *jitter*. Os demais métodos apresentam resultados próximos na maioria dos níveis de utilização do enlace principal.

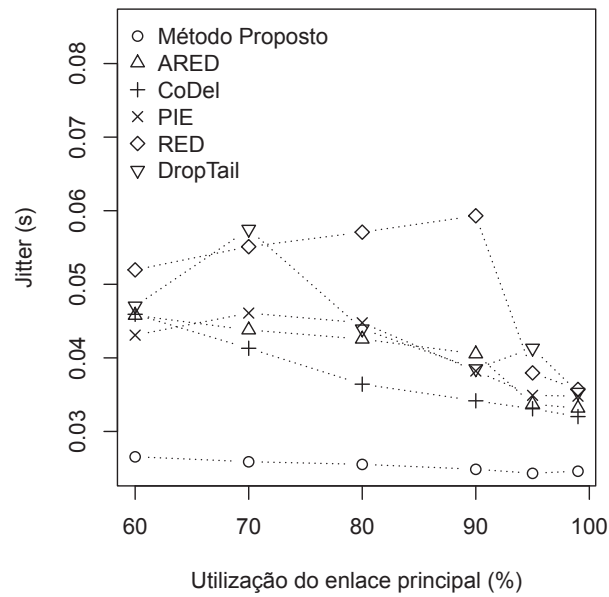


Figura 6.12: Jitter resultante para o vídeo *Sunflower*.

O resultado do atraso médio para o vídeo *Touchdown Pass* é apresentado pela Figura 6.13. Também observa-se o benefício da utilização do método proposto proporcionando menor atraso médio. O DropTail mantém o pior desempenho com maior atraso médio. O PIE, CoDel e ARED continuam com resultados intermediários de atraso médio.

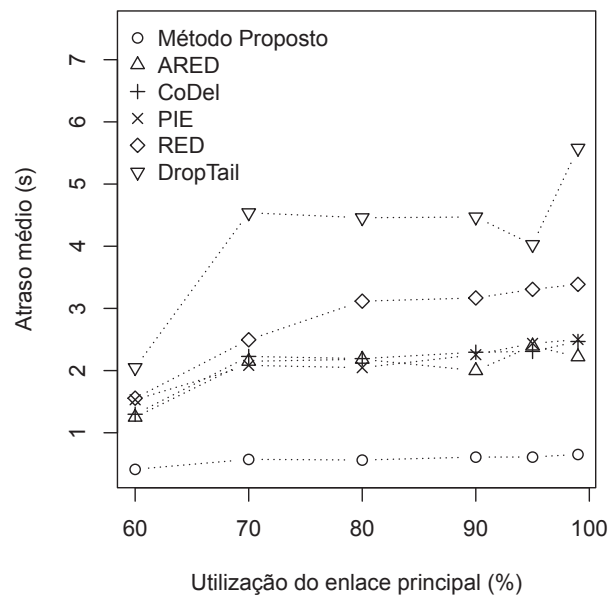


Figura 6.13: Atraso médio resultante para o vídeo *Touchdown Pass*.

Através da Figura 6.14 observa-se que o método AQM proposto também foi vantajoso na transmissão do vídeo *Touchdown Pass*, resultando em menor *jitter* para todos os níveis de utilização do enlace principal. O RED e o DropTail apresentam piores resultados com maior valor de *jitter* para quase todos os níveis de utilização do enlace principal. Os métodos AQM PIE, ARED e CoDel apresentam resultados intermediários.

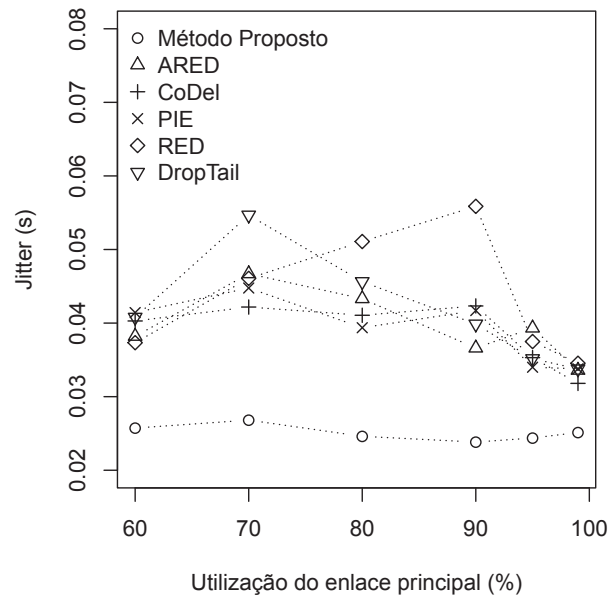


Figura 6.14: Jitter resultante para o vídeo *Touchdown Pass*.

Por meio da Figura 6.15 é possível observar o resultado da análise de atraso médio para o vídeo *Pedestrian Area*. Também fica claro o benefício da utilização do método proposto proporcionando menor atraso médio em todos os níveis de utilização. O DropTail mantém o pior desempenho com maior atraso médio para os níveis de utilização de 95% e 99% do enlace principal. O PIE, CoDel, RED e ARED continuam com resultados intermediários de atraso médio, com desvantagem para o PIE nos níveis de 80% e 99%.

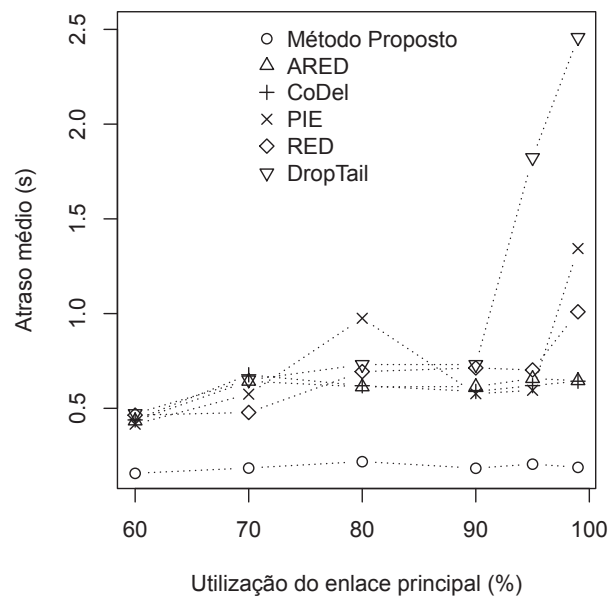


Figura 6.15: Atraso médio resultante para o vídeo *Pedestrian Area*.

Adicionalmente, na Figura 6.16 observa-se que o método AQM proposto também foi vantajoso na transmissão do vídeo *Pedestrian Area*, resultando em menor *jitter* para todos os níveis de utilização do enlace principal a partir de 60%. O RED, ARED, PIE e CoDel, apresentam resultados intermediários para quase todos os níveis de utilização, no qual o método CoDel obtém melhores resultados na partir de 90% de utilização do enlace principal. O DropTail apresentam

piores resultados com maior valor de *jitter* para quase todos os níveis de utilização do enlace principal a partir de 70%.

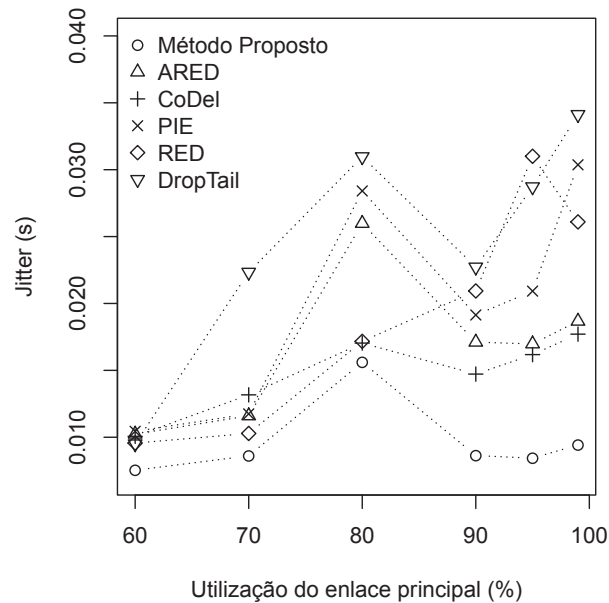


Figura 6.16: Jitter resultante para o vídeo *Pedestrian Area*.

O resultado do atraso médio para o vídeo *Riverbed* é apresentado pela Figura 6.17. Por meio dos resultados obtidos nos níveis de utilização a partir de 70% é possível constatar a vantagem do método proposto que tende a proporcionar menores valores para o atraso médio nos demais níveis de utilização. A partir de 80% de utilização observa-se a grande desvantagem do método DropTail que mantém elevados valores para o atraso médio. O PIE, CoDel, RED e ARED continuam com resultados intermediários de atraso médio, com exceção dos níveis de utilização de 80% e 95% em que o CoDel e o PIE proporcionam vantagem resultando em menores valores para o atraso médio.

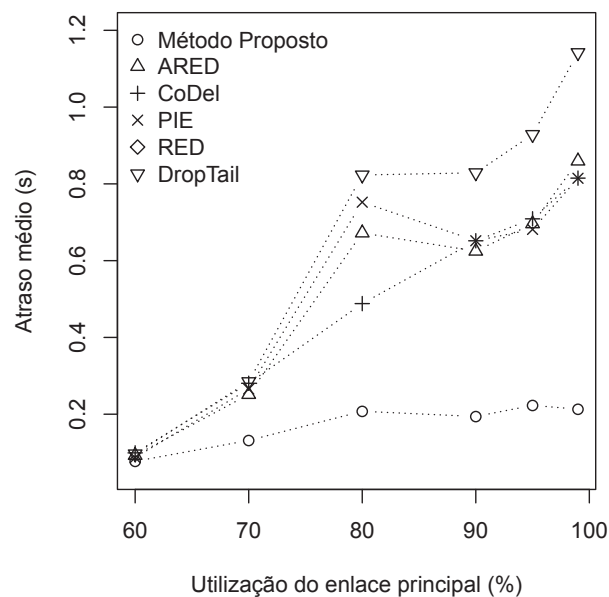


Figura 6.17: Atraso médio resultante para o vídeo *Riverbed*.

Finalizando, por meio da observação da Figura 6.18 nota-se que o método proposto também foi vantajoso na transmissão do vídeo *Riverbed*, resultando em menor *jitter* para todos os níveis de utilização do enlace principal a partir de 70%. O DropTail apresentou o pior resultado, com maior valor de *jitter* para todos os níveis de utilização do enlace principal a partir de 80%. Os métodos PIE, RED, ARED e CoDel apresentam resultados intermediários, com ligeira vantagem para o CoDel e o PIE, nos níveis de 80% e 95% de utilização.

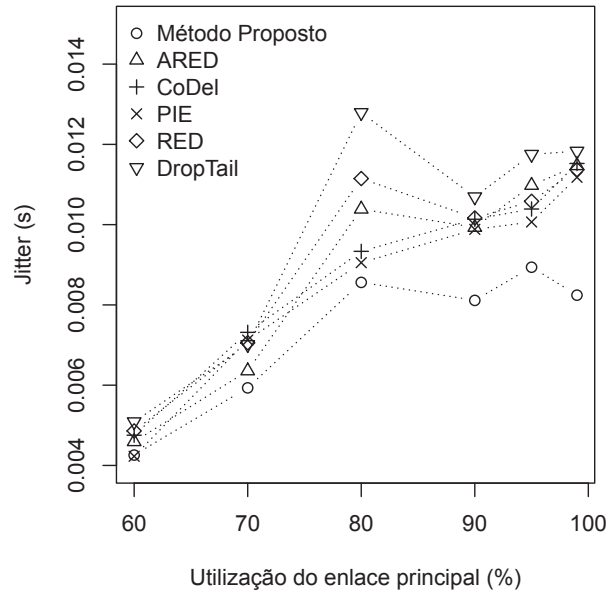


Figura 6.18: Jitter resultante para o vídeo *Riverbed*.

6.3 Justiça entre Fluxos

Para a análise da justiça, utilizou-se o índice de justiça de *Jain* [61]. O índice de justiça de *Jain* é um critério de desempenho utilizado em esquemas de alocação de recursos, bastante utilizado pela área de pesquisa de computação e telecomunicações. O índice de justiça de *Jain* não possui dimensão e pode assumir valores entre 0 e 1. Quanto mais próximo de 1, melhor a justiça. O cálculo do índice de justiça de *Jain* [46] é dado por:

$$\mathcal{J}(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^n x_i)^2}{n \cdot \sum_{i=1}^n x_i^2} \quad (6.1)$$

onde x_i corresponde aos valores dos resultados individuais a serem avaliados.

A Tabela 6.1 apresenta o índice de justiça de *Jain* para o *Throughput* para a transmissão do vídeo *Big Buck Bunny*, relacionando os métodos avaliados em função do nível de utilização do enlace principal, com a média calculada com intervalo de confiança de 95%.

Tabela 6.1: Índice de justiça de *Jain* para o *Throughput*.

Método	60%	70%	80%	90%	95%	99%	Média \pm h
DropTail	0.900	0.882	0.906	0.933	0.804	0.802	0.871 ± 0.058
RED	0.965	0.949	0.945	0.948	0.956	0.925	0.948 ± 0.013
ARED	0.947	0.955	0.974	0.964	0.954	0.943	0.956 ± 0.011
CoDel	0.988	0.931	0.907	0.923	0.960	0.968	0.946 ± 0.032
PIE	0.965	0.948	0.948	0.957	0.920	0.927	0.944 ± 0.018
Método Proposto	0.951	0.961	0.963	0.960	0.967	0.939	0.957 ± 0.010

Pela análise da tabela pode-se perceber que, de forma generalizada, o método proposto e o ARED tendem a promover maior justiça para os fluxos na maioria dos níveis de utilização do enlace principal. Porém, na média, o método proposto mantém um maior *throughput* em relação aos demais métodos avaliados. Nota-se também que o CoDel, apesar da queda no resultado em 80% e 90%, apresentou maior justiça em 99%. O DropTail obteve o pior desempenho em quase todos os níveis de utilização do enlace principal, promovendo menor justiça para os fluxos, principalmente em 95% e 99%. Os demais métodos apresentaram resultados intermediários.

A Tabela 6.2 apresenta o índice de justiça de *Jain* para os métodos avaliados em função do nível de utilização do enlace principal, com a média calculada com intervalo de confiança de 95%, para o atraso médio durante a transmissão do vídeo *Big Buck Bunny*.

Tabela 6.2: Índice de justiça de *Jain* para o atraso médio.

Método	60%	70%	80%	90%	95%	99%	Média \pm h
DropTail	0.771	0.903	0.653	0.616	0.680	0.550	0.695 ± 0.131
RED	0.887	0.691	0.674	0.837	0.597	0.853	0.757 ± 0.123
ARED	0.945	0.966	0.720	0.768	0.699	0.895	0.832 ± 0.123
CoDel	0.918	0.956	0.891	0.958	0.973	0.9362	0.939 ± 0.031
PIE	0.941	0.886	0.836	0.921	0.962	0.908	0.909 ± 0.046
Método Proposto	0.974	0.902	0.911	0.886	0.942	0.974	0.932 ± 0.039

Através da análise da Figura é possível observar que, na média, o CoDel e o método proposto mantêm, respectivamente, o primeiro e o segundo menor atraso médio para os fluxos em relação aos demais métodos avaliados. É interessante destacar que o método proposto promove maior justiça em 99%. Sendo assim, de maneira geral, pode-se constatar que o método proposto, CoDel e PIE promovem uma maior justiça aos fluxos em relação ao atraso médio. Percebe-se também que o DropTail apresenta os piores resultados, principalmente, a partir de 80% de utilização do enlace principal. Os demais métodos apresentaram resultados intermediários.

6.4 Transição entre Qualidades

Para realizar a análise de estabilidade da qualidade do vídeo no cliente, durante a transmissão do vídeo *Big Buck Bunny*, com nível de ocupação de 95%, foram capturadas as informações de segmentos de vídeo solicitados e respectivas taxas de qualidades. A Figura 6.19 apresenta a relação de segmentos e respectivas taxas (*bitrates*) que foram requeridos pelo cliente durante a transmissão do vídeo. Pela análise da figura nota-se claramente que o método proposto tende a manter uma maior estabilidade entre a transição de qualidades dos segmentos. Um menor número de transições consecutivas corresponde à uma transmissão mais estável, com suaves

alterações na qualidade. Essa suavização produz maior PSNR e melhora a justiça entre os fluxos, o que impacta diretamente na qualidade percebida pelo usuário.

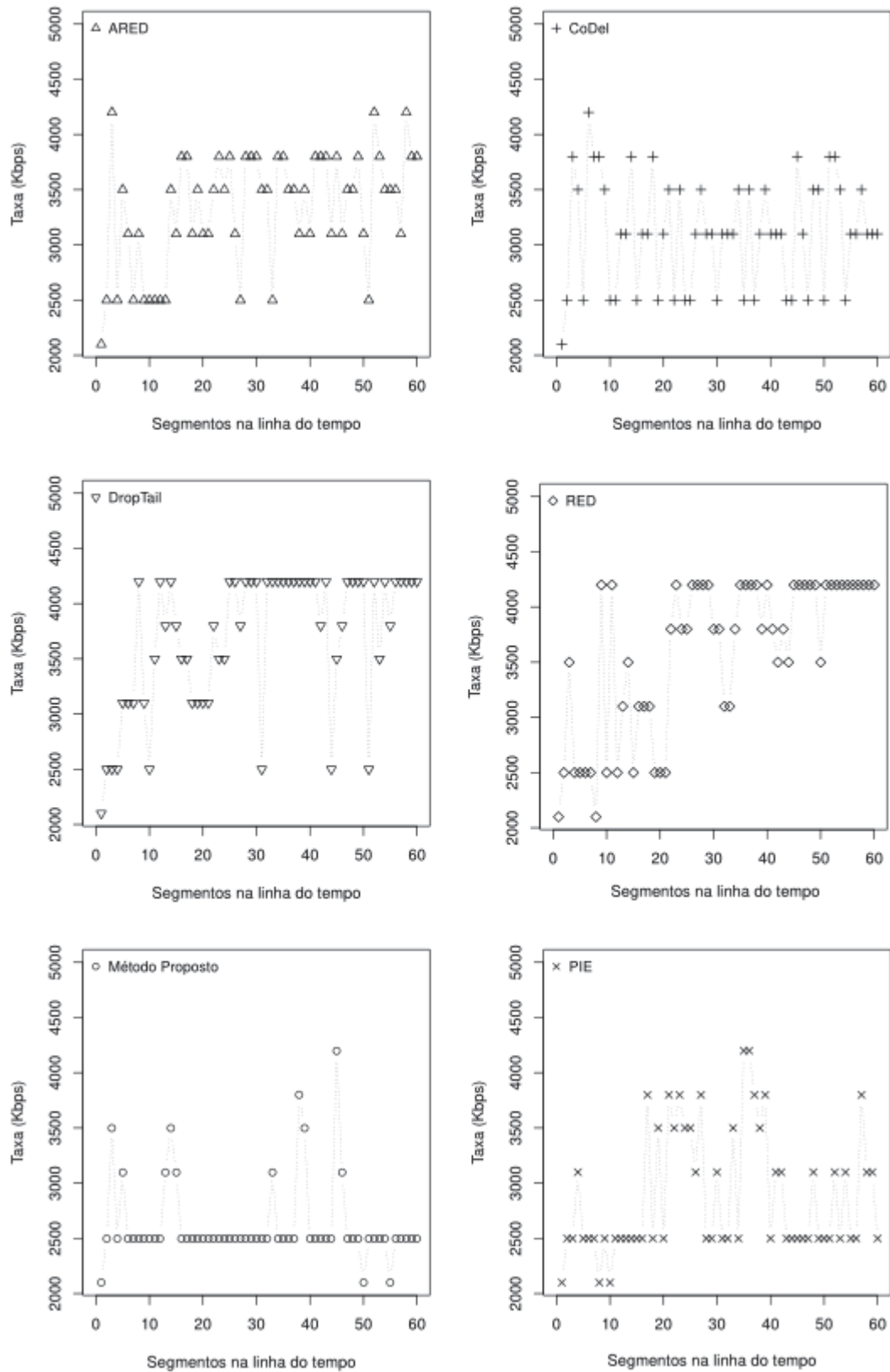


Figura 6.19: Transição entre os segmentos de vídeo solicitados pelo cliente para o vídeo *Big Buck Bunny* no nível de ocupação de 95%.

6.5 Discussão dos Resultados

Métodos AQM realizam descartes antecipados que notificam o servidor e cliente de vídeo bem como as fontes concorrentes, ao perceberem a perda de pacotes pelo TCP, sobre a necessidade de redução de suas taxas de transmissão. O TCP por sua vez efetiva a redução da taxa de transmissão por meio de seus algoritmos de controle de congestionamento, como *Slow Start* e o *Congestion Avoidance*. Com isso, o mecanismo de estimação do DASH, ao perceber que largura de banda disponível foi reduzida, seleciona os próximos segmentos de vídeo com menor qualidade, mantendo contínua a reprodução do conteúdo. O sistema DASH ao transferir o vídeo ao vivo em menor qualidade, faz com que o cliente perca menos segmentos, favorecendo a qualidade final do vídeo. De forma simplificada, os descartes de pacotes induzem a percepção de congestionamento, obrigando os fluxos DASH reduzirem a taxa de transmissão dos próximos segmentos. A transmissão dos segmentos de menor qualidade por sua vez possui menor quantidade de *bytes*. Consequentemente, há um melhor aproveitamento da capacidade de transmissão do enlace principal.

O PSNR é a relação entre a entrada e a saída de um processo de compressão com perdas, que avalia o quanto a compressão introduziu ruídos na imagem ou quadro original. Desta forma, quanto maior o valor do PSNR, significa melhor qualidade. Utilizando o PSNR podemos ter um valor numérico que representa a qualidade geral do vídeo. De maneira geral, observa-se que para moderados níveis de utilização do enlace principal, como em 60% e 70%, não há uma diferença expressiva entre o PSNR médio obtido pelos métodos avaliados. Contudo, ao aumentar o nível de utilização a partir de 80%, os métodos AQM atuam de forma a descartar pacotes antecipadamente, cada um utilizando seus respectivos mecanismos de estimação e detecção do nível de congestionamento, além das características particulares de seleção e descarte de pacotes, o que impulsiona diferentes valores de PSNR médio para diferentes métodos avaliados.

Observa-se também, de forma genérica, que o DropTail não é benéfico para o fluxo de vídeo adaptativo. A partir de 80% de utilização do enlace principal, o valor do PSNR médio tende a cair bastante em relação aos obtidos pelos demais métodos. Pode-se constatar que a queda no valor do PSNR médio a partir do nível de utilização 80% independente do vídeo transmitido. Consequentemente, observou-se o efeito negativo resultante em pacotes de vídeo que aguardam em longas filas para serem reproduzidos no cliente, além do aumento do atraso médio e jitter. Pode-se concluir, em menor escala, que isso tem relação com o fenômeno do *Bufferbloat*.

É importante mencionar que durante a transmissão de alguns dos vídeos, como *Rush Hour* e *Sunflower*, em níveis de utilização de 70%, 80% e 90%, alguns dos métodos concorrentes se sobressaíram em relação ao desempenho do método proposto. Isso evidencia a principal fraqueza do método proposto, que possui um comportamento muito mais agressivo que os demais, descartando mais pacotes de forma precoce, o que pode ser melhorado diminuindo a quantidade de descartes em níveis de utilização baixos e intermediários.

Através da avaliação de desempenho, foi possível constatar que o método proposto é promissor, mas para ser estável deve tornar-se menos agressivo, principalmente nos momentos em que os níveis de utilização das filas ainda não chegam à 95%. Isso pode ser aperfeiçoado por meio de um ajuste dinâmico na probabilidade de descarte, reduzindo essa probabilidade quando os níveis de congestionamento ainda não são observados pelo controlador SDN.

Os resultados apresentaram ganhos significativos na utilização do método proposto para vídeos com maior compactação espacial e temporal, como no vídeo *Big Buck Bunny*. É interessante observar que o vídeo *Big Buck Bunny* possui o menor valor para *Motion Activity* dentre todos os vídeos avaliados, que justifica o resultado favorável em termos de PSNR. A perda de segmentos em vídeos menos compactados é muito mais severa, como nos vídeos *Pedestrian*

Area e *Riverbed*, que possuem os maiores valores para *Motion Activity* dentre os demais vídeos avaliados.

Ao analisar os diferentes tipos de conteúdo transmitidos em relação às diferentes técnicas AQM, observa-se que o vídeo de maior duração, *Big Buck Bunny*, tende a favorecer maior valor de PSNR médio, pois a perda de alguns poucos segmentos de vídeo não tem grande impacto no resultado final da média. Já em vídeos de curta duração, como o *Riverbed*, percebe-se que por menor que seja o congestionamento, já há impactos no PSNR médio, uma vez que são menos quadros envolvidos para realizar a comparação com o vídeo original.

Com exceção dos resultados da transmissão do vídeo *Sunflower*, percebe-se nos demais resultados, que o método CoDel é o grande concorrente do método proposto, uma vez que apresenta resultados bastante próximos em quase todos os níveis de utilização do sistema. Pode-se sugerir que uma alteração dinâmica mais agressiva nos parâmetros *Interval* e *Target* otimizaria a política de descarte e aumentaria a eficiência do algoritmo, que muitas vezes se demonstra bastante comedido, como nos níveis de 95% e 99% de utilização do enlace principal.

O método proposto obtém melhor desempenho para fluxos DASH se comparados aos demais métodos avaliados, porém, de forma geral, os resultados só indicam que esse ganho é efetivo a partir do nível de utilização de 95% no enlace principal. Observa-se então que o uso do método proposto satisfaz o seu objetivo, no qual visa apresentar melhores resultados quando considerando o congestionamento em níveis críticos. Em situações de congestionamento, a partir dos 95% de utilização do enlace principal, a melhoria da qualidade de vídeo, a redução do atraso médio e *jitter*, nos dá indicativos da eficiência de atuação do mecanismo de controle de congestionamento na camada de rede, evidenciada com a redução da taxa de envio de pacotes pelo nó de origem com a finalidade de se reduzir o congestionamento da rede como um todo.

Em relação à transição entre as qualidades dos segmentos de vídeo, considerando os demais métodos AQM avaliados, percebe-se que o método proposto tende a manter uma maior estabilidade entre a transição de qualidades dos segmentos. Essa estabilidade reflete diretamente na qualidade percebida pelo usuário.

Apesar dos resultados favoráveis ao método proposto nas situações de maior congestionamento, fica evidente que o controle da probabilidade de descarte de pacotes deve ser aprimorado, de forma que o método proposto possibilite melhores resultados também em níveis de baixa utilização do sistema. Porém, vale mencionar que a proposta não somente favorece o fluxo de vídeo adaptativo como também auxilia no escoamento do tráfego originado nas fontes concorrentes, possibilitando um menor atraso médio e um menor *jitter* para a entrega dos pacotes aos destinos.

Capítulo 7

Conclusão

Operadoras aumentam o tamanho das filas em seus roteadores, como forma de acomodar o tráfego, gerando o fenômeno *Bufferbloat*. Ficando por longos períodos enfileirados, os pacotes de vídeo acabam perdendo a validade para serem reproduzidos no cliente. Mecanismos de AQM se tornaram importantes reguladores de congestionamentos e justiça no compartilhamento da largura de banda entre as aplicações. Dessa forma, algoritmos de gerenciamento ativo de filas podem ser utilizados como importantes reguladores de congestionamentos, de forma a cooperar com o algoritmo de estimação e adaptação de conteúdo utilizado pela tecnologia DASH.

Na entrega de vídeos dinamicamente adaptados em tempo real a quantidade de segmentos a serem entregues ao cliente dentro de um limite tolerável de latência tem impacto significativo na qualidade do vídeo assistido. O atraso e a perda de segmentos pode gerar períodos de congelamento na imagem pela falta de segmentos a serem reproduzidos, caso a rede passe por congestionamentos, afetando negativamente a qualidade da imagem percebida.

Este trabalho explorou como o tráfego DASH interage com os mecanismos de AQM, propondo um método que foca em aplicações de fluxos de vídeo ao vivo sendo transmitidas em enlaces com elevados níveis de utilização. Foi implementado um método para gerenciamento ativo de filas com descarte aleatório randômico de pacotes, baseado no tempo estimado de permanência do pacote na fila do roteador. O modelo utiliza como principal paradigma o limite temporal em que um pacote de vídeo permanece na fila, o que consequentemente impacta na probabilidade com que este pacote poderá ser descartado, promovendo maior eficiência para aplicações sensíveis ao atraso, como é o caso do *streaming* de vídeo ao vivo.

O estudo também apresentou uma revisão sobre transmissão de vídeos adaptativos com DASH e métodos de gerenciamento ativo de filas. Os experimentos foram realizados por simulações de transmissão reais de vídeos. Os resultados são comparados aos métodos apresentados na revisão. Os resultados obtidos para diferentes tipos de vídeo, transmitidos em um cenário relativamente realista sobre uma topologia *dumbbell*, mostram que a proposta é promissora, especialmente quando aplicada para vídeos que apresentem maior compactação espacial e temporal ou quando o enlace de gargalo na rede tem um nível de ocupação bastante alto.

O uso de métodos AQM para melhorar a eficiência dos roteadores nas transmissões de conteúdo de vídeo tem reflexos profundos sobre o desempenho da rede de acesso e sobre a qualidade percebida pelo usuário. O uso do método AQM proposto permite que o cliente se adapte antes do congestionamento, possibilitando que o reproduzidor reaja de forma rápida e robusta às mudanças da rede, resultando em maior PSNR médio para enlaces com elevados níveis de utilização. Um ponto forte da proposta é a simplicidade computacional. Um ponto fraco do método proposto é a necessidade de configuração dos parâmetros t_{min} e t_{max} .

7.1 Trabalhos Futuros

Trabalhos futuros envolvem a avaliação da capacidade de adaptação do vídeo utilizando diferentes resoluções e taxas de quadros por segundo, além das representações em qualidades distintas. Mecanismos podem ser propostos para realizar a configuração automática dos parâmetros t_{min} e t_{max} , buscando um maior desempenho a ser obtido pelo ajuste dinâmico dos limiares independente de um controlador SDN, fazendo com que o algoritmo tenha autonomia para decidir sobre a alteração dos valores.

Outro ponto a ser investigado é a política de atualização da probabilidade de descarte de pacotes p_i que pode obter maiores ganhos utilizando um ajuste através de um mecanismo de média móvel exponencial juntamente com fatores de incremento e decremento, como utilizado do *PIE*, para proporcionar ajustes dinâmicos de acordo com a latência média na fila detectada. Uma nova melhoria a ser analisada é a implementação de um temporizador para permissão para rajadas de pacotes, proporcionando um breve período sem descartes, como implementado no CoDel.

Estudos complementares abrangem a ampliação do cenário visando fornecer uma gama maior de experimentos em topologias com mais roteadores, além de diversos clientes de vídeo, de maneira a fornecer resultados que demonstrassem o desempenho da proposta para outras configurações de parâmetros. Em especial, o tamanho da fila e a capacidade dos enlaces podem ser variados. Dessa forma será possível realizar uma maior variação de testes para constatar a os ganhos para outras amostras de vídeos e realizar uma avaliação detalhada sobre os diferentes tipos de cenários. Sendo assim a avaliação pode ser complementada trazendo resultados diferentes de forma a revelar outras perspectivas sobre o desempenho da proposta.

REFERÊNCIAS

- [1] Patrick Seeling and Martin Reisslein. Video traffic characteristics of modern encoding standards: H.264/AVC with SVC and MVC extensions and H.265/HEVC. *The Scientific World Journal*, 2014, 02.
- [2] J. Kua and G. Armitage. Optimising DASH over AQM-Enabled Gateways Using Intra-Chunk Parallel Retrieval (Chunklets). In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9, July 2017.
- [3] Dinesh C. Verma, Hui Zhang, and Domenico Ferrari. Delay Jitter Control for Real-Time Communication in a Packet Switching Network. 09 2001.
- [4] Cisco Systems. Cisco VNI: Forecast and Methodology, 2016–2021. White paper: The Zettabyte Era: Trends and Analysis. Technical report, Cisco Systems, USA, 2016.
- [5] Anatoliy Zabrovskiy, Evgeny Petrov, Evgeny Kuzmin, and Christian Timmerer. Evaluation of the Performance of Adaptive HTTP Streaming Systems. *CoRR*, abs/1710.02459, 2017.
- [6] Jason Greengrass, John Evans, and Ali C. Begen. Not All Packets Are Equal, Part 2: The Impact of Network Packet Loss on Video Quality. *IEEE Internet Computing*, 13:74–82, March 2009.
- [7] Xiph.org Derf’s Video Test Media Collection, YUV HD Content and Full Sequences - <https://media.xiph.org/video/derf/>, 2018.
- [8] Y. Li, X. Gong, W. Wang, X. Que, and J. Ma. An Autonomic Active Queue Management Mechanism to Improve Multimedia Flow Delivery Quality. In *2010 International Conference on Communications and Mobile Computing*, volume 1, pages 493–497, April 2010.
- [9] Thomas Stockhammer. Dynamic Adaptive Streaming over HTTP –: Standards and Design Principles. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems, MMSys ’11*, pages 133–144, New York, NY, USA, 2011. ACM.
- [10] E. Grigorescu, C. Kulatunga, and G. Fairhurst. Evaluation of the Impact of Packet Drops due to AQM over Capacity Limited Paths. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–6, Oct 2013.
- [11] Jim Gettys and Kathleen Nichols. Bufferbloat: Dark Buffers in the Internet. volume 9, New York, NY, USA, 11 2011. ACM.
- [12] Yang Xiaogang, Jiqiang Liu, and Li Ning. Congestion Control Based on Priority Drop for H.264/SVC. In *MUE.2007.107*, pages 585 – 589, 2007.

- [13] R. Adams. Active Queue Management: A Survey. *IEEE Communications Surveys Tutorials*, First Quarter:1425–1476, 2013.
- [14] Sally Floyd and Van Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, pages 397–413, August 1993.
- [15] Subramanya P, Vinayaka Ks, Gururaj H L, and Ramesh B. Performance evaluation of high speed tcp variants in dumbbell network. *IOSR Journal of Computer Engineering*, 16:48–53, 01 2014.
- [16] S. Patel, P. Gupta, and G. Singh. Performance measure of Drop tail and RED algorithm. In *2010 2nd International Conference on Electronic Computer Technology*, pages 35–38, May 2010.
- [17] Sally Floyd, Ramakrishna Gummadi, and Scott Shenker. Adaptive RED: An Algorithm for Increasing the Robustness of RED’s Active Queue Management. *AT&T Center for Internet Research at ICSI*, 2001.
- [18] Kathleen Nichols and Van Jacobson. Controlling Queue Delay. volume 10, pages 20:20–20:34, New York, NY, USA, May 2012. ACM.
- [19] Rong Pan, Preethi Natarajan, Chiara Piglion, Mythili Suryanarayana Prabhu, Vijay Subramanian, Fred Baker, and Bill Versteeg. PIE: A Lightweight Control Scheme to Address the Bufferbloat Problem. 07 2013.
- [20] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin. A self-configuring RED gateway. In *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*, volume 3, pages 1320–1328, March 1999.
- [21] Jacobson V. Nichols K. Controlled Delay Active Queue Management. RFC 8289 (Proposed Standard), 2018.
- [22] I. Järvinen and M. Kojo. Evaluating CoDel, PIE, and HRED AQM techniques with load transients. In *39th Annual IEEE Conference on Local Computer Networks*, pages 159–167, Sep. 2014.
- [23] Gopal Thapa. Video Compression Techniques: A Survey. 09 2010.
- [24] A. Tamhankar and K. R. Rao. An overview of H.264/MPEG-4 Part 10. In *Proceedings EC-VIP-MC 2003. 4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications (IEEE Cat. No.03EX667)*, volume 1, pages 1–51 vol.1, July 2003.
- [25] I. Sodagar. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia*, 18(4):62–67, April 2011.
- [26] Douglas E.Comer and David L.Stevens, editors. *Internetworking with TCP/IP, Volume III, Client Server Programming and Applications*. Prentice Hall, 1996.
- [27] N. Bouten, M. Claeys, S. Latré, J. Famaey, W. Van Leekwijck, and F. De Turck. Deadline-Based Approach for Improving Delivery of SVC-Based HTTP Adaptive Streaming Content. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–7, May 2014.

- [28] Stefan Lederer, Christopher Müller, and Christian Timmerer. Dynamic Adaptive Streaming over HTTP Dataset. In *Proceedings of the 3rd Multimedia Systems Conference, MMSys '12*, pages 89–94, New York, NY, USA, 2012. ACM.
- [29] C. Concolato, J. Le Feuvre, F. Denoual, F. Mazé, E. Nassor, N. Ouedraogo, and J. Taquet. Adaptive Streaming of HEVC Tiled Videos Using MPEG-DASH. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8):1981–1992, Aug 2018.
- [30] J. Kua, G. Armitage, and P. Branch. The Impact of Active Queue Management on DASH-Based Content Delivery. In *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, pages 121–128, Nov 2016.
- [31] J. Kua, G. Armitage, and P. Branch. A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming over HTTP. *IEEE Communications Surveys Tutorials*, 19(3):1842–1866, thirdquarter 2017.
- [32] Jean Le Feuvre, Cyril Concolato, Nassima Bouzakaria, and Viet-Thanh-Trung Nguyen. MPEG-DASH for Low Latency and Hybrid Streaming Services. In *Proceedings of the 23rd ACM International Conference on Multimedia, MM '15*, pages 751–752, New York, NY, USA, 2015.
- [33] C. Concolato, J. Le Feuvre, F. Denoual, F. Mazé, E. Nassor, N. Ouedraogo, and J. Taquet. Adaptive Streaming of HEVC Tiled Videos Using MPEG-DASH. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8):1981–1992, Aug 2018.
- [34] T. Lohmar, T. Einarsson, P. Fröjdh, F. Gabin, and M. Kampmann. Dynamic adaptive HTTP streaming of live content. In *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–8, June 2011.
- [35] Stefan Lederer, Christopher Müller, and Christian Timmerer. Dynamic Adaptive Streaming over HTTP Dataset. In *Proceedings of the 3rd Multimedia Systems Conference*, pages 89–94, New York, NY, USA, 2012. ACM.
- [36] J. Le Feuvre, J-M. Thiesse, M. Parmentier, M. Raulet, and C. Daguet. Ultra High Definition HEVC DASH Data Set. In *Proceedings of the 5th ACM Multimedia Systems Conference, MMSys '14*, pages 7–12, New York, NY, USA, 2014. ACM.
- [37] Christian Kreuzberger, Daniel Posch, and Hermann Hellwagner. A Scalable Video Coding Dataset and Toolchain for Dynamic Adaptive Streaming over HTTP. In *Proceedings of the 6th ACM Multimedia Systems Conference*, pages 213–218, New York, NY, USA, 2015. ACM.
- [38] Jason J. Quinlan, Ahmed H. Zahran, and Cormac J. Sreenan. Datasets for Evaluation of Dynamic Adaptive Streaming over HTTP. In *Proceedings of the 7th International Conference on Multimedia Systems*, New York, NY, USA, 2016. ACM.
- [39] S. D. Patil and M. P. Tahiliani. On the robustness of AQM mechanisms against non-responsive traffic. In *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6, Nov 2016.
- [40] H. J. Kim, P. K. Park, H. S. Yoon, and S. G. Choi. QoS-aware Active Queue Management scheme for multimedia services. In *13th International Conference on Advanced Communication Technology (ICACT2011)*, pages 1037–1042, Feb 2011.

- [41] P. Taht H. Joergensen, T. Mc Kenney. The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm. *RFC*, 8290, 2018.
- [42] M. Shreedhar and G. Varghese. Efficient Fair Queuing using Deficit Round-Robin. *IEEE/ACM Transactions on Networking*, 4(3):375–385, June 1996.
- [43] Telecommunication Standardization Sector of ITU. Methodology for Subjective Assessment of the Quality of Television Pictures. Recommendation ITU-R BT.500-13, January 2012.
- [44] D.P. Torres and E.M.G. Fernandez and E.P. Ribeiro and V.A. Reguera and C. de Oliveira. On the impact of adaptive RED in IP networks transporting H.264/MPEG-4 AVC video streams. *Computers and Electrical Engineering*, 38(1):128 – 139, 2012.
- [45] George F. Riley and Thomas R. Henderson. *The NS-3 Network Simulator - Modeling and Tools for Network Simulation*, pages 15–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [46] Book review: The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling by raj jain (john wiley & sons 1991). *SIGMETRICS Perform. Eval. Rev.*, 19(2):5–11, September 1991. Reviewer-Al-Jaar, Robert Y.
- [47] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, 103(1):14–76, Jan 2015.
- [48] Jirka Klaue, Berthold Rathke, and Adam Wolisz. EvalVid - A Framework for Video Transmission and Quality Evaluation. *Lecture notes in computer science*, 2794:255–272, 03 2004.
- [49] S. Jeannin and A. Divakaran. Mpeg-7 visual motion descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):720–724, June 2001.
- [50] Dong Kwon Park, Yoon Seok Jeon, and Chee Sun Won. Efficient use of local edge histogram descriptor. In *Proceedings of the 2000 ACM Workshops on Multimedia*, MULTIMEDIA '00, pages 51–54, New York, NY, USA, 2000. ACM.
- [51] M. R. Zakerinasab and M. Wang. Dependency-aware distributed video transcoding in the cloud. In *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, pages 245–252, October 2015.
- [52] FFmpeg Developers. FFmpeg Multimedia Framework - <https://www.ffmpeg.org>.
- [53] J. Sauer C. Silva, G. Souza and C. Pedroso. Evaluation of Impairment Caused by MPEG Video Frame Loss. *Engineering*, pages 493–503, 2017.
- [54] Jean Le Feuvre, Cyril Concolato, and Jean-Claude Moissinac. GPAC: Open Source Multimedia Framework. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, pages 1009–1012, New York, NY, USA, 2007.
- [55] I. Sodagar. The MPEG-DASH Standard for Multimedia Streaming over the Internet. *IEEE MultiMedia*, 18(4):62–67, April 2011.

- [56] Doreid Ammar, Thomas Begin, and Isabelle Guerin-Lassous. A New Tool for Generating Realistic Internet Traffic in NS-3. In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, pages 81–83, Brussels, Belgium, 2011.
- [57] Nelson L. S. Fonseca, Gilberto S. Mayor, and Cesar A. V. Neto. On the equivalent bandwidth of self-similar sources. *ACM Trans. Model. Comput. Simul.*, 10(2):104–124, April 2000.
- [58] R. Serral-Gracià, E. Cerqueira, M. Curado, M. Yannuzzi, E. Monteiro, and X. Masip-Bruin. An Overview of Quality of Experience Measurement Challenges for Video Applications in IP Networks. In *Proceedings of the 8th International Conference WWIC*, Berlin, Heidelberg, 2010. Springer-Verlag.
- [59] Alex Junior Figueiredo Torres, Eduardo Parente Ribeiro, and Carlos Marcelo Pedroso. Predictive Delay-centric Handover for Video Streaming over SCTP. *Computer Communication*, 87:49–59, August 2016.
- [60] J. Greengrass, J. Evans, and A. C. Begen. Not All Packets Are Equal, Part I: Streaming Video Coding and SLA Requirements. *IEEE Internet Computing*, 13(1):70–75, Jan 2009.
- [61] Raj Jain, Dah Ming Chiu, and Hawe WR. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *CoRR*, cs.NI/9809099, 01 1998.